ROBUST MOTION PLANNING IN PRESENCE OF UNCERTAINTIES USING A MANEUVER AUTOMATON

A Thesis
Presented to
The Academic Faculty

by

Julide J. Topsakal

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

School of Aerospace Engineering
Georgia Institute of Technology
April 2005

ROBUST MOTION PLANNING IN PRESENCE OF UNCERTAINTIES USING A MANEUVER AUTOMATON

Approved by:

Prof. Carlo L. Bottasso, Chair,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Prof. John R. Olds,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Prof. Panagiotis Tsiotras,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Date Approved: 15 April 2005

To my family and all my friends

for their endless encouragement and trust,

In memory of my friend Yann Michaël Ricou,

deceased December 3rd, 2004

ACKNOWLEDGEMENTS

Nothing can be compared to the undefinite, unconditional and unrelenting love parents share with their children. Therefore I would first like to express my gratitude towards my parents, Mehmet and Béatrice Topsakal. I will never be able to thank both of them enough, for standing by me and pushing me to do my best, for showing me what is the most important in life. Without their active presence in my life, guidance, motivation, courage and willingness to teach me everything in their power, I would have never come this far. Thank you for being so convincing, powerful and just. I owe everything that I have accomplished and everything that I am to the best choices you made for me and the best path you showed me.

My sister, Merih Topsakal, is the best example I have in life. She is the most rational, the most hardworking and the most sensitive and loving person I know. I would like to thank her for sharing all those endless qualities with me.

I would like to devote a few words of gratitude to my grandmother, Granny, who was a teenager in France during the Second World War and therefore was incapable of continuing her education. In her eyes, I am the accomplishment of her never fulfilled dream. Thank you for helping me through hard times and for having a nice word of encouragement whenever I needed.

As far as my personal development is concerned, there are numerous people I would like to acknowledge. First amongst those are the best friends of my parents, the Köymen and Ergün families. Thank you for making our family an extended one. I will always remember the road trips we took together because they have helped me realize the strong link between humans, science and nature. Their impact is considerable when I set my priorities. I would also like to thank my best friends and friends Aylin Göktürk, Abdoulkadir Rais, Okan Edis and many others for making high school unforgettable and for backing me up and encouraging me all these years. Without their friendship I would have not been this

courageous. I am also particularly grateful to Engin Satana, who, besides my parents and sister, has the biggest impact in who I am today. I would like to thank him for mimicking all the moods I have been through, it be good or bad, being beside me whenever I was lost. I would like to recognize the fun moments I spent with Ayca Ozkur, Elif Ucuz and Megan Parrish. Needless to say that I owe most of my accomplishments here at Georgia Tech to the friends I have acquainted with here. They are the ones who helped me most balance academic and personal life. In particular, I would like to acknowledge Sebastian Leibold, Florian Zickfeld, Joscha Hofmann. Special thanks to my two unforgettable roommates, brothers Daniel Contesse and Felipe Orfali for being available 24 hours a day. Their passion, their intelligence, their smiles and the absolute friendship they devoted to me is what reminded me why I was here. I cannot forget mentioning the encouragement I received from my labmates, Luca Riviello, Nazmiye Acikgöz, Faik Sümer and Chong Seok Chang and from my classmates and friends, Adrian Koller, Venkatesh Madyastha, Nimrod Rooz, and Yoko Watannabe. I have learned as much through motivating discussions with all of them as I have with any faculty member. Therefore, I would like to thank all of them for their suggestions and time. Last but not for the least, I would like to thank Julien Ricordeau, who is the reason why this last semester at Tech will remain an unforgettable one. I am forever indebted to you for your understanding, endless patience, trust, support, help and encouragement when it was most required. Merci beaucoup pour tout.

This research relies on realistic data, thanks to the simulations performed on the GT-Max. I am grateful to Suresh K. Kannan, for his help in setting up the experiments run on the simulator.

I would like to acknowledge everyone who played a role in my education, advised me and supported me with all critical decisions. Thank you for showing me the right way to go and for backing me up and cheering me each time I thought I was not up to the task. I would especially like to thank the academic faculty of the School of Aerospace Engineering for making the education at Tech challenging, interesting and rewarding. I am grateful to Prof. John Olds and Prof. Panagiotis Tsiotras for their willingness to participate in my advisory committee.

Finally and foremost, I would like to thank my advisor, Dr. Carlo L. Bottasso, who exposed me to the world of autonomous flight. It was the greatest pleasure to work with you. I feel very privileged to have been one of your students. Thank you for your trust and inspiring ideas, for the time you devoted to me and for involving me in your research. What I have learned from you is irreplaceable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

MA	Maneuver Automaton
ODE	Ordinary Differential Equation
UAV	Unmanned Aerial Vehicle
$\boldsymbol{c}_{eq}(\boldsymbol{x}, \boldsymbol{u})$	Denotes equality constraints on states and controls
$\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u})$	Denotes inequality constraints on states and controls
$\dot{(\bullet)} = \mathrm{d}(\bullet)/\mathrm{d}t$	Derivative with respect to time
J	Cost function
J^R	Robust cost function
J_{opt}	Optimal cost function for nominal optimization
J^R_{opt}	Optimal robust cost function for worst case optimization
n_T	length of $\boldsymbol{q}$
n_M	length of $\boldsymbol{p}$
p	Maneuver
$\boldsymbol{p}$	Vector of trajectory maneuvers
$\mathscr{P}_M$	Set of all maneuvers defining the library
q	Trim condition
q_{from}	Trim condition a maneuver is initiated from
q_{to}	Trim condition a maneuver leads to
$\boldsymbol{q}$	Vector of trajectory trims
$\mathscr{Q}_T$	Set of all relative equilibria defining the library
$\boldsymbol{q}_{opt}$	Optimal vector of trajectory trims for nominal optimization
$\boldsymbol{q}^R_{opt}$	Optimal vector of trajectory trims for worse case policy
r	Yaw rate while in trim
ϵ_r	Uncertainty in the trim yaw rate

S	Invariant mechanical system
t	Time
t_0	Initial time
t_f	Final time
T	Fixed maneuver duration
ϵ_T	Uncertainty on the fixed maneuver duration
$\boldsymbol{T}$	Vector of trajectory maneuver durations
$\boldsymbol{u}$	Denotes the control inputs to S
$\mathscr{U}_T$	Set of all possible sequences of motion primitives
u_{fwd}	Forward velocity while in trim
u_{side}	Side velocity while in trim
$\epsilon_{u_{fwd}}$	Uncertainty in the forward trim velocity
$\epsilon_{u_{side}}$	Uncertainty in the side trim velocity
ϵ_r	Uncertainty in the yaw rate r
$\varepsilon_{u_{\mathrm{fwd}}}$	Percent error in the forward trim velocity
$\varepsilon_{u_{\mathrm{side}}}$	Percent error in the side trim velocity
$\varepsilon_{u_{\mathrm{r}}}$	Percent error in the trim yaw rate
$\boldsymbol{w}$	Generalized velocities
$\boldsymbol{w}_q$	Generalized velocities specific to trim q
$\epsilon_{\boldsymbol{w}_q}$	Uncertainties on the generalized velocities specific to trim q
X	State space of S
$\boldsymbol{x}$	Denotes the states
$\boldsymbol{x}_i$	Initial states
$\boldsymbol{x}_f$	Final states
$\Delta\boldsymbol{x}_p$	Fixed maneuver displacement
$\epsilon_{\Delta\boldsymbol{x}_p}$	Uncertainty on the fixed maneuver displacement
τ	Variable trim coasting time
$\boldsymbol{\tau}$	Vector of trajectory trim coasting times
$\boldsymbol{\tau}_{opt}$	Optimal vector of trajectory trim coasting times for nominal optimization

$\boldsymbol{\tau}_{opt}^{R}$	Optimal vector of trajectory trim coasting times for worst case policy
$\phi(\cdot)$	Function describing the evolution of the system while in trim
γ	Uncertainty level
Δt_c	Time increment between two consecutive controller activations in receding horizon

SUMMARY

The scope of this research is to develop an algorithm that enables an autonomous vehicle to construct a trajectory between an initial position and a final determined target. The trajectory is executed minimizing a performance index and thus, solving an optimal control problem. The discretization of the system dynamics is favored by the use of a hybrid maneuver automaton, whose characteristics are stored within a motion primitive library. To bring robustness to the overall behavior of the vehicle, uncertainties are accounted for. Those uncertainties in the dynamics of the vehicle are directly introduced within the motion primitive library. Georgia Tech's autonomous GT-Max simulator is used in order to obtain all the data provided in this library.

The data acquired represents the dynamics of the system within the maneuver automaton framework and is used to build a graph of possible trajectories, given initial conditions on the dynamics of the system and on the position vector. A nominal optimal control problem is run for each trajectory, omitting uncertainties. In parallel, an optimization which incorporates the uncertainties on the motion primitives within the motion planning problem is carried on. Feedback is introduced in a model predictive way, by the use of Receding Horizon Control. Consequently, the problem is repeated for both optimizations in closed loop until the system reaches the final target within an allowable tolerance margin.

Considering a minimum time problem, it can be predicted that, although the nominal trajectory is initially the fastest one, it performs very poorly in closed-loop when uncertainties are introduced. The reason for this poor behavior is the fact that, the solutions to the nominal optimization problem involve maneuvers, which are associated with uncertainties large in magnitude. The closed-loop trajectory of the policy in which uncertainties are accounted for is expected to perform better and to be less perturbed given that it uses more cautious primitives.

CHAPTER I

INTRODUCTION

This research proposes a new formulation of the motion planning problem for unmanned aerial vehicles. Recent research efforts in this area have relied on deterministic models. Uncertainties were only accounted for during trajectory tracking, which results in the correction of model uncertainties only a posteriori, while the whole movement planning is done in a purely deterministic fashion. This is fallacious because it relies on the assumption that all uncertainties can be reduced to disturbances of deterministically planned trajectories. It is therefore essential to introduce uncertainties both at the path planning and trajectory tracking levels. The algorithm that was developed for this research seeks to incorporate uncertainties in vehicle parameters at the level of motion planning.

1.1 Hierarchical Decomposition of the Flight Control System

The motion planning problem of autonomous vehicles is complex in nature and therefore computational tractability is mostly difficult to achieve. In order to reduce this complexity autonomous vehicles undergo a hierarchical decomposition of controls into decision layers [25]. More precisely, three layers are defined:

Strategic layer: This layer corresponds to the definition of the mission objectives by a central command decision making entity, in most cases a human operator.

Tactical layer: In this layer the motion planning algorithm decides on how to best fulfill the goals set by the upper strategic layer.

Reflexive layer: This layer, also called skill layer, consists in tracking the trajectory planned at the tactical layer in addition to regulating and stabilizing the vehicle.

This hierarchical decomposition mimics the way human piloted aircrafts operate. For example, in such a system, the strategic layer will correspond to the pilot receiving a mission and preflight briefings from the control tower, the tactical layer matches the decision process the pilot undergoes on how to best accomplish the mission given to him in accordance with the briefings provided, and finally the mission will depend on the pilot's actual skills.
In current Unmanned Aerial Vehicles (UAV) the higher layer in the hierarchy is still given to the hands of humans. However, it is desired to achieve complete autonomous behavior of the two other layers. The proposed research will concentrate on adding robustness to the tactical layer. The motion planning will be performed using a maneuver automaton (MA) representation of the dynamics of the vehicle.

1.2 Maneuver Automaton

At the tactical level, the motion planning problem for UAVs is usually tackled as an optimal control problem. In fact, simple solutions such as purely kinematic planning or definition of way-points and subsequent spline interpolation will lead, in general, to unfeasible trajectories for highly agile autonomous vehicles. Recent effort in this area take into full account the systems' dynamic characteristics [12, 15, 23]. The MA representation of the dynamics of the system is a modelling language for UAV trajectories, that is based on the interconnection of motion primitives, each primitive corresponding to a feasible and therefore trackable vehicle maneuver. The optimal control problem of path planning is then transcribed into the simpler problem of finding the best sequence of motion primitives within the library that minimizes a given cost function, which in the case presented is time. The path planning equations presented in Pb. 5.1 are very time consuming, and therefore costly, if integrated in real-time. In order to achieve computational tractability, the MA framework was introduced, in which the discretization is done off-line.

1.3 Dealing with Uncertainties

In complex real-life applications, especially in hostile environments, there are important sources of uncertainty that affect the decision making of autonomous agents. The uncertainties affecting a system can be classified as

- epistemic, i.e. due to a lack of knowledge, and
- aleatory, also known as objective or inherent uncertainty.

The inevitable presence of uncertainties is usually relegated at the level of the reflexive layer, which then takes the additional role of making the system properly robust to model disturbances. This implies that the motion planning problem is usually regarded as a purely deterministic one, while the stochastic nature of the vehicle control is dealt with only at the lower reflexive layer. It is this inner real-time control loop that ensures proper tracking of the planned trajectory, eliminating the effects of uncertainties on the knowledge of the vehicle physical characteristics and of the environment. In the MA representation of the dynamics as introduced in [12, 15], uncertainty is mainly present in the parameters of the motion primitives. These are due to poor modelling of the dynamics of the system as well as to external environmental changes. At the level of motion planning, a feasible sequence of motion primitives as a result of an optimization problem will lead, in the presence of uncertainties to perturbed and even unfeasible trajectories. Therefore, an optimization based on the nominal policy where uncertainties are disregarded is not optimal.

1.4 Research Motivation

The decision making process that takes place at the tactical level is not and should not be seen as a purely deterministic operation, but should account for the presence of uncertainties. In fact, a capable pilot operating at the tactical level makes tactical decisions that maximize the chances of reaching the mission objectives. These decisions are based on a number of complex non-deterministic interacting factors, such as the operator's inevitably limited information, willingness to assume a certain level of risk, ability to estimate the probable reward implied by a given action, skill in assessing the risks implied by a given

decision, and ultimately on the operator's experience in predicting the outcome of given decisions in uncertain situations. By using purely deterministic processes at the tactical level, one negates the fundamental risk-management nature of the human tactical decision making process.

The uncertainties in system parameters and in the environment are essential and cannot be simply reduced to disturbances of deterministically planned paths. Failure to incorporate epistemic uncertainty also at the level of the tactical layer can lead to mission failure, since the reflexive layer is not capable of correcting wrong decisions made at the tactical layer. In order to better explain this concept, consider for example the case of path planning in the presence of multiple moving dynamic obstacles, as for example in the case of multiple agents co-operating on a common goal, but acting independently and making autonomous decisions. A deterministically computed trajectory could fail to avoid an obstacle and lead to a potential crash, for example because the relative positions of the vehicles were only approximatively known. Given the fatal trajectory, no reflexive skills can avoid the crash, since the lower level controller can only blindly follow the prescribed path. In other words, wrong policies can not be compensated by the sole use of control skills.

A part from the epistemic uncertainties, other processes that affect the operations of UAVs are aleatory in nature, and can not be reduced to disturbances. For example, an enemy can engage a UAV only with a given probability. Therefore, the decision to prefer, say, a shorter but more dangerous path over a longer but safer one can only be taken in a risk-sensitive way: the risk of being detected has to be weighted against the probable rewards implied by the more direct path. The tactical decisions made by experienced human operators are heavily influenced by these risk-management policies. This mode of operation should be reflected by autonomous control systems for UAVs making tactical decisions. However, the currently adopted deterministic approach at the tactical layer negates this view and, consequently, is inherently limited in its ability to conduct effective autonomous missions in highly complex, hostile environments. Accounting for uncertainties in the path planning level, we are able to

- mimic to a higher level of fidelity the decision making process performed by human

pilots,

- produce trajectories that maximize the probability of meeting the goals set forth at the strategic level, weighing risks against performance,
- guarantee greater overall robustness in the control system.

1.5 Thesis Outline

This thesis will be organized as follows:

- First, the Maneuver Automaton representation of the vehicle dynamics will be presented in detail, which will also include the motion primitive library that was defined for the research. In addition, information on how the uncertainty library was build is provided.
- In chapter III, the library of motion primitives will be constructed,
- In chapter IV, the robust optimal control theory will be introduced
- General motion planning and motion planning with the Maneuver Automaton framework will be defined.
- Next, the nominal and robust optimization approaches will be introduced. In addition, the closed loop formulation will be presented.
- In chapter VII, results to both optimal control problems will be provided.
- Finally, conclusions will be drawn, problems encountered identified and future improvements and investigations to be conducted discussed.

CHAPTER II

MANEUVER AUTOMATON

At the source of the Maneuver Automaton lies the hybrid automata. In order to better explain the motivation under the MA framework, the reader will be given a brief overview of hybrid systems and hybrid automata. Furthermore, the MA representation of the dynamics of a system will be defined.

2.1 Hybrid Systems and Hybrid Automata

2.1.1 Hybrid Systems

A hybrid system is, per definition, a mixed discrete-continuous system [7]. It consists of a discrete program within an analog environment. Generally speaking, such systems originate whenever one mixes logical decision-making with the generation of continuous control laws, such as in modern flight control systems.

The continuous dynamics of the system are usually represented by differential equations of the form

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \qquad \forall t \in \mathbb{R}^+$$

where $\boldsymbol{x}(t) \in \mathcal{X}$ is the continuous component of the state and $f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is a controlled vector field depending on the continuous state $\boldsymbol{x}$ and controls $\boldsymbol{u}$. The controls $\boldsymbol{u}$ consist of a continuous component and a discrete phenomenon and therefore are referred to as hybrid. Discrete controller events can include autonomous switching or jumps or controlled switching or jumps. Continuous time systems are also affected by discrete events, such as jumps in state.

2.1.2 Hybrid Automata

More and more processes are controlled by programs which are rooted in a continuously changing environment. Therefore, those programs must be designed such that they adapt to

the changes in the environment in real-time. However, continuously changing environments are usually approximated by discrete sampling. The hybrid automata theory was developed in order to represent both discrete and continuous processes within a unified framework. More precisely, a hybrid automata is a finite-state model for the dynamics of a system possessing both discrete and continuous components and has been proposed as a formal model for hybrid systems (see [17, 1, 2, 22] for more detail).

An example of a hybrid system taken from [17] would be a digital controller of an analog plant. In hybrid automata theory [17], the discrete state of the digital controller is modelled by the vertices and the discrete dynamics of the controller is modeled by the edges of a directed graph. The continuous dynamics of the plant are modeled by ordinary differential equations. The execution of a hybrid automata results in a continuous change (flows) and discrete change (jumps). For more insight, we recall the following definition from [17, 1, 2, 22]

Definition 2.1 (Hybrid Automata)**.** *A hybrid automaton consists of*

Variables: *A finite set* $X = \{x_1, \ldots, x_n\}$ *with* $x_i \in \mathbb{R}$ *,* $i = [1, \ldots, n]$ *and* $n = dim(X)$*.* $\dot{X} = \{\dot{x}_1, \ldots, \dot{x}_n\}$ *is the finite set of first derivatives during continuous change and* $X' = \{x'_1, \ldots, x'_n\}$ *is the finite set of values at the conclusion of discrete change.*

Control graph: *A finite directed multigraph* (V, E) *where vertices* $\in V$ *are called control modes and edges* $\in E$ *are called control switches.*

Initial, invariant and flow conditions: *Three properties are assigned to each control mode: the initial, the invariant and the flow conditions.*

Jump condition: *Property assigned to each control switch.*

Event: *Property assigned to each control switch.*

As an example, temperature control of a thermostat, modeled by a hybrid automata is provided in Fig. 1.

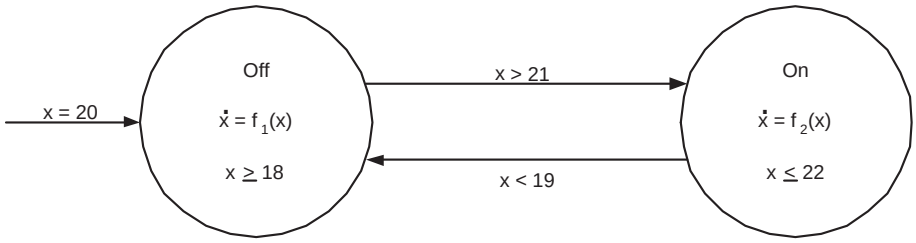


Figure 1: Hybrid automata model of a thermostat

2.2 Maneuver Automaton

The Maneuver Automaton modeling of the dynamics of the system was introduced by Frazzoli in [12, 15]. A MA is defined as a new computational and modeling framework for steering underactuated, nonholonomic mechanical systems such as mobile robots and autonomous vehicles that are time-invariant and admit symmetries and relative equilibria. It is based on a quantization of the system's dynamics, by which the feasible nominal system trajectories are restricted to the family of curves that can be obtained by the interconnection of suitably defined primitives. These primitives then constitute a library of motion or trajectory primitives which is at the core of the MA.

Within the MA framework, the control architecture involves switching from one trajectory primitive to the other, always alternating between two classes of primitives, which are defined next. The control system therefore includes both continuous and discrete dynamics and thus is referred to as a hybrid controller. Moreover, the state of the MA model is represented by a hybrid vector, as it takes a hybrid control input, which represents feasible trajectory primitives for the system.

2.3 Definition of Motion Primitives

A maneuver automaton applies to a class of nonlinear mechanical control systems admitting symmetries and relative equilibria. Under that assumption, a motion primitive, also referred to as trajectory primitive, is an equivalence class of trajectories, as defined in [12]. With the introduction of trajectory primitives, the relevant features of the dynamics of the system in hand are captured. Furthermore, good approximations of optimal solutions are

obtained by interconnecting appropriate primitives.

Motion primitives are composed of two classes, trim and maneuver conditions. At each instant in time, the vehicle is either performing a maneuver or is within a trim condition. They were defined considering a mechanical control system on a Lie group and using Lie algebra. The reader is referred to [8] and to App. A for more information on simple mechanical control systems on Lie groups.

The advantage of a Lie group framework is that it leads to coordinate free expressions of the behavior of the system as well as controls. Hence, one can exploit local charts and use Lie Group framework to move on the configuration space without reformulating the controls.

2.3.1 Trim Conditions

The first class of motion primitives defined are trim trajectories which correspond to dynamic and relative equilibria conditions. More precisely, trim conditions are characterized by steady-state trajectories, in which the velocity $\boldsymbol{v}$ of the vehicle expressed in the body frame as well as the control setting $\boldsymbol{u}$ are constants. Hovering and constant rate turning flight at constant speed are both examples of trim trajectories.

While in a trim condition q, a vehicle can coast for an unlimited amount of time, referred to as coasting time and denoted as τ in this work. τ constitutes the continuous component of the hybrid control vector of the MA representation.

2.3.2 Maneuver Conditions

A maneuver is a finite time, fixed displacement transition from one relative equilibria, i.e. trim condition, to the next one. Notice that this does not omit the possibility of transitioning from and back to the same trim trajectory. Each maneuver p is characterized by

- its finite duration T specific to each maneuver,
- its fixed displacement $\Delta\boldsymbol{x}_p$,
- its fixed heading change $\Delta\psi$.

The fixed displacement specific to each maneuver represents the discrete component of the hybrid control vector. The trim condition after which the maneuver is initiated will be

denoted as q_{from} and the trim condition the maneuver leads to as q_{to}.

The maneuver is initiated by the control action p for a specific maneuver. This control action is a function of q_{from} and can only lead to q_{to}.

2.4 Motion Primitive Library

For the scope of this research, the MA framework will be used to produce a library by selecting a finite number of relative equilibria, connected by a finite number of maneuvers. Let $\mathscr{Q}_T$ be the set of all possible trims selected and $\mathscr{P}_M$ be the corresponding set of associated maneuvers. The number of maneuvers in $\mathscr{P}_M$ is variable and will be decided according to the problem in hand. All trims and maneuvers selected will be depicted by a directed graph. An example for a simple problem is presented in Fig. 2.

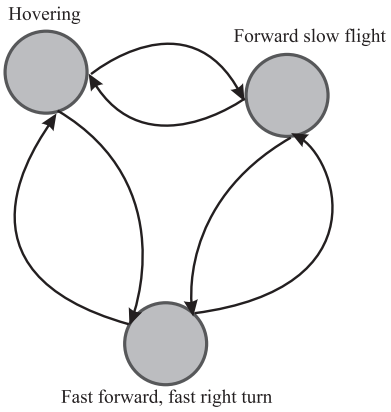


Figure 2: Simple Maneuver Automaton

2.4.1 Trim conditions

As mentioned earlier, each trim is characterized by a constant velocity in body axes. Consequently, at each vertex, the body axis velocity components will be stored. As the problem in consideration is two dimensional, the body axis velocities will be the forward and sideways velocities as well as turning rates. A trim condition will then include, within the library:

- u_{fwd}, its forward velocity on the body x-axis,
- u_{side}, its sideways velocity on the body y-axis,
- r, its yaw rate.

We also will denote $\boldsymbol{v} = (u_{\text{fwd}}, u_{\text{side}}, r)^T$ as the velocity vector in the body attached frame. The uncertainty within a trim condition comes from the gap between the theoretical velocities stored in the library vertices and the real values for those velocities. These uncertainties will also be stored in the library vertices. Notice that while on the graph vertex, the discrete state is stationary while the continuous dynamics keep evolving. In other words, in a trim condition the dynamics are defined as

$$\boldsymbol{x} = \phi(\boldsymbol{w}\tau)\boldsymbol{x}_i \tag{1}$$

where $\boldsymbol{w}$ represent the generalized velocities and $\boldsymbol{x}_i$ is the initial state at which the trim starts. $\phi(\boldsymbol{w}\tau)$ describes the evolution of the system within a trim and is an exponential [12].

2.4.2 Maneuvers

The edges of the directed graph correspond to the transitions between two trim states, i.e. maneuvers within the library. At each edge will be stored information proper to maneuvers which include

- the trim condition the maneuver issues from q_{from},
- the trim condition the maneuver leads to q_{to},
- finite transition time T,
- fixed displacement $\Delta\boldsymbol{x}_p$,
- fixed heading change $\Delta\psi$.

The uncertain parameters will also be added to the library and will include the uncertainty in the displacement $\Delta\boldsymbol{x}_p$, in the heading change $\epsilon_{\Delta\psi}$ and in the fixed maneuver duration T.

2.5 Building the Reference Library

For the scope of this research, different libraries of motion primitives were used in order to obtain numerous numerical results. Trims defined include fast and slow forward flights, fast

and slow steady left and right turns. Tables 1 through 4 summarize the trim and maneuver conditions that were defined for this investigation. It must be pointed out at this point that the libraries used are very limited ones, since they do not encompass the entire flight envelope. The vehicle is indeed constrained to a very limited amount of possible actions, motion primitives, which it is allowed to perform. Thus, optimal trajectories obtained to an optimization problem will constitute suboptimal ones. In order to have a more general library and gain in optimality, one can add as many trim conditions and maneuvers.

For example, consider a motion planning problem where a system is required to reach a final position in minimum time (general motion planning problems will be defined in detail in later sections). Let us also assume that the system is given an initial heading such that it has to turn in order to accomplish a mission. It is trivial in this case that if the library of motion primitives was enriched with an on the spot turn, a possible feasible solution to an optimization problem will be a trajectory in which the system rotates to the desired heading and flies straight. This way, it does not have to perform a turn that requires spacial displacement. Angular displacement while keeping its spacial position might represent a better optimal solution. This problem is addressed in [12] in more detail.

In the scope of this research, we will be limited to the choice of trims and maneuvers shown in Table 1 through Table 4. Also notice that the fixed duration and fixed displacement of maneuvers is not added to the library at this point. The next sections will deal with how to add those values to the library.

2.5.1 First Reference Library

The below presented Table 1 and Table 2 were preliminary trim and maneuver conditions that were defined in order to validate the proposed approach. In directed graph representation, those trims and maneuvers are connected as in Fig. 3 As seen, trim conditions include fast and slow steady forward flights and fast and slow steady turns.

Table 1: Reference Trim Library - Kinematics based

Trim q	u_{fwd} [m/s]	u_{side} [m/s]	r [rad/s]	Description
1	1	0	0	slow forward flight
2	4	0	0	fast forward flight
3	1	0	-0.1	slow left turning flight with slow turn rate
4	1	0	0.1	slow right turning flight with slow turn rate
5	4	0	-0.4	fast left turning flight with fast turn rate
6	4	0	0.4	fast right turning flight with fast turn rate

Table 2: Reference Maneuver Library - Kinematics based

Maneuvers p	q_{from}	q_{to}	Maneuvers p	q_{from}	q_{to}
1	2	1	10	5	2
2	1	2	11	2	6
3	3	1	12	6	2
4	1	3	13	6	5
5	1	4	14	5	6
6	4	1	15	5	3
7	3	4	16	3	5
8	4	3	17	6	4
9	2	5	18	4	6

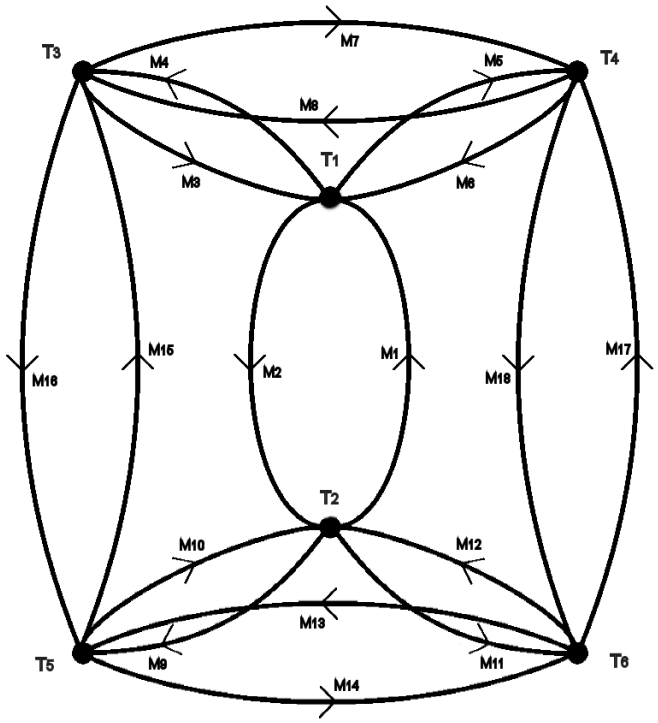


Figure 3: Directed Graph of the Maneuver Automaton - Kinematics based

2.5.2 Reference Library obtained through simulations on the GTMax

Table 3 and Table 4 provide the initial maneuver automaton library that is used, in the following section, to evaluate uncertainties present in motion primitives using the GTMax simulator.

In directed graph representation, those trims and maneuvers are connected as shown in Fig. 4

Table 3: Reference Trim Library - Experiment based

Trim q	u_{fwd} [ft/s]	u_{side} [ft/s]	r [rad/s]	Description
1	0	0	0	hover
2	3	0	0	slow forward flight
3	10	0	0	fast forward flight
4	3	0	-0.1	slow left turning flight with slow turn rate
5	3	0	0.1	slow right turning flight with slow turn rate
6	10	0	-0.6	fast left turning flight with fast turn rate
7	10	0	0.6	fast right turning flight with fast turn rate

Table 4: Reference Maneuver Library - Experiment based

Maneuvers p	q_{from}	q_{to}	Maneuvers p	q_{from}	q_{to}
1	2	1	12	5	4
2	1	2	13	6	3
3	1	3	14	3	6
4	3	1	15	3	3
5	3	2	16	7	7
6	2	3	17	6	7
7	4	2	18	7	6
8	2	4	19	7	5
9	2	5	20	5	7
10	5	2	21	6	4
11	4	5	22	4	6

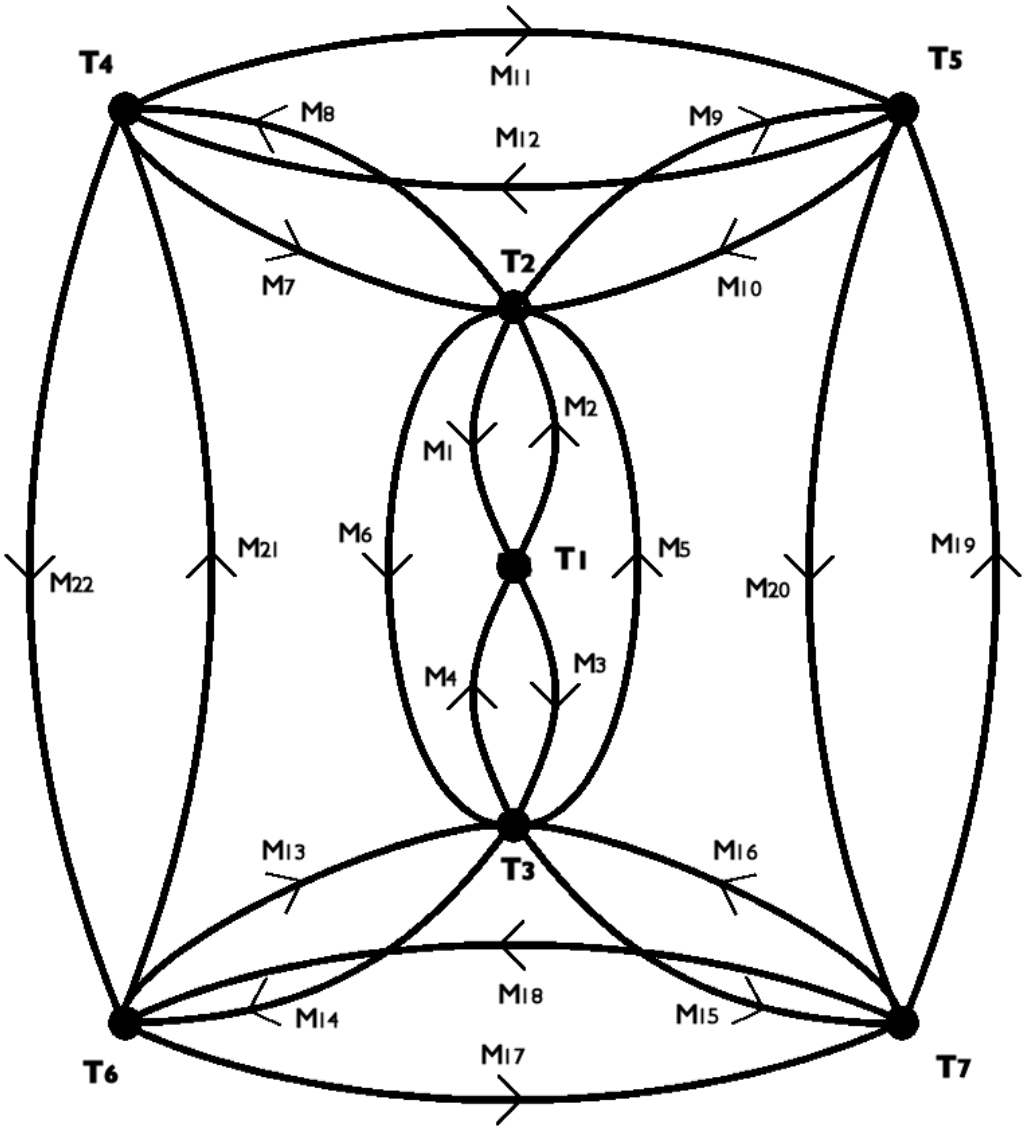


Figure 4: Directed Graph of the Maneuver Automaton - Experiment based

2.6 Populating the Library

By populating the library we mean finding the characteristics of the above chosen maneuver conditions in the Maneuver Automaton representation of the dynamics of our system. We will again use two different approaches in evaluating maneuver characteristics given the two preliminary libraries defined above. One library will be enlarged using kinematics based evaluation while the second one will be obtained through flight simulations.

2.6.1 Populating the First Reference Library

This first reference library was built using physics. In fact, one can predict, using physics and kinematics, what displacement is required and how long those transitions last given appropriate assumptions. For example, it is easy to predict the required displacement, heading change and duration of a transition between two straight flights if one assumes known constant acceleration. We based our analysis of maneuver characteristics on the following: it was assumed that maneuver durations were known. Table 5 shows the durations that were used. Now using average acceleration required to steer the system from

Table 5: Maneuver Library with durations - Kinematics based

p	q_{from}	q_{to}	T [s]	p	q_{from}	q_{to}	T [s]
1	2	1	2	10	5	2	4
2	1	2	2	11	2	6	4
3	3	1	0.2	12	6	2	4
4	1	3	0.2	13	6	5	4
5	1	4	0.2	14	5	6	4
6	4	1	0.2	15	5	3	4
7	3	4	0.2	16	3	5	4
8	4	3	0.2	17	6	4	4
9	2	5	4	18	4	6	4

one relative equilibria to another, we can compute the required displacement and heading changes using the expression for the average acceleration

$$\bar{\boldsymbol{a}} = \frac{\boldsymbol{v}_{q_{\text{from}}} - \boldsymbol{v}_{q_{\text{to}}}}{T}$$

where $\boldsymbol{v} = (v_{\text{fwd}}, v_{\text{side}}, r)^T$ expressed in body attached frame and $\bar{\boldsymbol{a}} = (\bar{a}_{\text{fwd}}, \bar{a}_{\text{side}}, \bar{\alpha})^T$. The fixed linear maneuver displacement is then obtained writing

$$\Delta \boldsymbol{x}_p = \frac{1}{2}\,\bar{\boldsymbol{a}}\,T^2 + \boldsymbol{v}_{q_{\text{from}}} T$$

Here $\Delta \boldsymbol{x}_p$ represents the fixed displacement relative to the local body frame. Notice that the third component of the displacement vector gives the angular displacement required to reach the final yaw rate. We will isolate that equation from $\Delta \boldsymbol{x}_p$ and express the angular displacement $\Delta\psi$ as

$$\Delta\psi = \frac{1}{2}\,\bar{\alpha}\,T^2 + r_{q_{\text{from}}} T$$

Then by projecting the components of $\boldsymbol{v}_{q_{\text{to}}}$ onto the body frame attached to the point where the maneuver is initiated, we obtain a new expression of $\Delta \boldsymbol{x}_p$ by setting the linear average acceleration to

$$\bar{\boldsymbol{a}} = \frac{1}{T}\begin{bmatrix} v_{\text{to}}^{\text{fwd}} \cos\Delta\psi - v_{\text{from}}^{\text{fwd}} \\ v_{\text{to}}^{\text{fwd}} \sin\Delta\psi \end{bmatrix}$$

Those equations apply only under the assumption that we can locally treat the problem as linear and that the side velocities in the library are all zero. This results in the enlargement of the library since information on fixed displacement and fixed heading changes becomes available. The resulting Maneuver Automaton is presented in Table 6.

Table 6: Maneuver Library with displacement and heading change - Kinematics based

p	q_{from}	q_{to}	T [s]	ϵ_T [s]	$\Delta\boldsymbol{x}_{p_x}$ [m]	$\Delta\boldsymbol{x}_{p_y}$ [m]	$\Delta\psi$ [rads]	$\epsilon_{\Delta\boldsymbol{x}_{p_x}}$ [m]	$\epsilon_{\Delta\boldsymbol{x}_{p_y}}$ [m]	$\epsilon_{\Delta\psi}$ [rads]
1	2	1	2.0	-0.5	5.0	0.0	0.0	-0.5	0.0	0.0
2	1	2	2.0	-0.5	5.0	0.0	0.0	-0.5	0.0	0.0
3	3	1	0.2	-0.05	0.2	-0.05	$-\frac{2\pi}{180}$	-0.05	0.0	0.0
4	1	3	0.2	-0.05	0.2	-0.05	$-\frac{2\pi}{180}$	-0.05	0.0	0.0
5	1	4	0.2	-0.05	0.2	0.05	$\frac{2\pi}{180}$	-0.05	0.0	0.0
6	4	1	0.2	-0.05	0.2	0.05	$\frac{2\pi}{180}$	-0.05	0.0	0.0
7	3	4	0.2	-0.05	0.2	0.0	0.0	-0.05	0.0	0.0
8	4	3	0.2	-0.05	0.2	0.0	0.0	-0.05	0.0	0.0
9	2	5	4.0	-0.5	14.0	-10.0	$-\frac{45\pi}{180}$	-7.0	5.0	$-\frac{5\pi}{180}$
10	5	2	4.0	-0.5	14.0	-10.0	$-\frac{45\pi}{180}$	-7.0	5.0	$-\frac{5\pi}{180}$
11	2	6	4.0	-0.5	14.0	10.0	$\frac{45\pi}{180}$	-7.0	-5.0	$\frac{5\pi}{180}$
12	6	2	4.0	-0.5	14.0	10.0	$\frac{45\pi}{180}$	-7.0	-5.0	$\frac{5\pi}{180}$
13	6	5	4.0	-0.5	14.0	0.0	0.0	-7.0	5.0	$\frac{5\pi}{180}$
14	5	6	4.0	-0.5	14.0	0.0	0.0	-7.0	5.0	$\frac{5\pi}{180}$
15	5	3	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
16	3	5	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
17	6	4	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
18	4	6	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$

The procedure described here was mainly formulated in order to validate the robust motion planning problem that will be introduced in the upcoming sections. It constitutes a valid approach in sketching an academic example however would not give very good results in practice since it does not rely on a mathematical model of the vehicle nor on data from a real hardware. Therefore, the values of the various quantities characterizing motion primitives computed would be very questionable for a real vehicle. Libraries of motion primitives can be build more accurately taking into account the dynamics of the system in question. We discuss two main ways to acquire information on maneuver and trim conditions.

1. Experiments could be conducted either by simulation or by using the real hardware. Those would consist in simulating sequences of trims and maneuvers using a simulator or, in case a real vehicle model is available, having a pilot fly the vehicle in order to get the recorded data [23].

2. Libraries can be built by optimization. Maneuver information can be obtained off-line solving corresponding optimal control problems specific to each maneuver, based on a mathematical or flight mechanics model of the vehicle [6]. Given a performance index to minimize or maximize, suitable constraints and bounds on the states and controls, one solves a maneuver optimal control problem whose solution yields the control time histories that fly the vehicle. Hence, maneuvers can be defined in a mathematically clear way. The maneuver optimal control problem solved consists of a boundary value problem whose solution is expensive to compute. Therefore transciption methods are used to compute the solution to the discretized problem. In other words, first one discretizes the differential equations and next solve the discrete optimal control problem [5].

In this work, maneuver information and extension of the motion primitive library is rendered possible by the use of a flight tested simulation tool: the GT-Max.

2.6.2 Populating the Reference Library through Simulations

Information on maneuver conditions is very difficult to obtain. In fact all the previously described enlargement of the library is based on the assumption that the problem of finding the displacement and heading change between two relative equilibria can locally be treated as linear. Moreover, it was assumed that the maneuver durations were known. One can have numerous different ways to transition from one trim condition to the other. For example considering a maneuver condition connecting two forward flights at different speeds, the transition can be treated at constant known acceleration. Then knowing the required velocity change will provide the required duration and displacement.

It is therefore difficult to obtain, physically, rigorous information on what assumptions to use. Also, maneuvers depend on the dynamics of the system in use, as the they change drastically if the system in question is a hovercraft or an aircraft. It was therefore of interest to investigate the nature of those maneuvers and how they affect the overall behavior of the system on a real time flight test verified simulator.

CHAPTER III

MOTION PRIMITIVE LIBRARY

3.1 Maneuver Automaton Representation

We will use the two previously introduced maneuver automaton representation of the dynamics of our system in this section.

3.2 Evaluating the Uncertainties

3.2.1 Kinematic Evaluation

In the previous section, the library of motion primitives was enlarged, by kinematic evaluation, to contain information on the required displacement and heading change, while the vehicle is maneuvering from one relative equilibria to another. Similarly, uncertainties present on maneuver characteristics are evaluated in this chapter. We assumed for our analysis that the uncertainty on trim velocities were known. The uncertainties present in trim characteristics as defined in Section 2.3 were modeled as percent errors from theoretical values. For example, for straight flights, it was assumed that faster straight flights will suffer more from external perturbations, such as environmental uncertainty, than slower flights. For turning flights, it was assumed that the uncertainties, such as environmental perturbations, will affect more faster turns than it will affect slower turns. Hence,

- the percent error on forward velocity u_{fwd} denoted as $\varepsilon_{u_{\text{fwd}}}$,
- the percent error on side velocity u_{side} denoted as $\varepsilon_{u_{\text{side}}}$,
- the percent error on yaw rate r denoted as ε_r.

were added to Table 7. Lets assume that the uncertainties relative to maneuver durations are known. Since we also know the uncertainty present in trim velocities, we can repeat the analysis from Section 2.6.1 including uncertain terms. Hence, we include the resulting errors on maneuver displacements and maneuver heading changes in Table 8.

Table 7: Trim Library with errors on velocities - Kinematics based

q	u_{fwd} [m/s]	u_{side} [m/s]	r [rad/s]	$\varepsilon_{u_{\text{fwd}}}$	$\varepsilon_{u_{\text{side}}}$	ε_r
1	1.0	0.0	0.0	10 %	0 %	0 %
2	4.0	0.0	0.0	20 %	0 %	0 %
3	1.0	0.0	-0.1	10 %	0 %	10 %
4	1.0	0.0	0.1	10 %	0 %	10 %
5	4.0	0.0	-0.4	15 %	0 %	20 %
6	4.0	0.0	0.4	15 %	0 %	20 %

Table 8: Maneuver Library with errors on displacement and heading change - Kinematics based

p	q_{from}	q_{to}	T [s]	ϵ_T [s]	$\Delta \boldsymbol{x}_{p_x}$ [ft]	$\Delta \boldsymbol{x}_{p_y}$ [ft]	$\Delta\psi$ [rads]	$\epsilon_{\Delta \boldsymbol{x}_{p_x}}$ [ft]	$\epsilon_{\Delta \boldsymbol{x}_{p_y}}$ [ft]	$\epsilon_{\Delta\psi}$ [rads]
1	2	1	2.0	-0.5	5.0	0.0	0.0	-0.5	0.0	0.0
2	1	2	2.0	-0.5	5.0	0.0	0.0	-0.5	0.0	0.0
3	3	1	0.2	-0.05	0.2	-0.05	$-\frac{2\pi}{180}$	-0.05	0.0	0.0
4	1	3	0.2	-0.05	0.2	-0.05	$-\frac{2\pi}{180}$	-0.05	0.0	0.0
5	1	4	0.2	-0.05	0.2	0.05	$\frac{2\pi}{180}$	-0.05	0.0	0.0
6	4	1	0.2	-0.05	0.2	0.05	$\frac{2\pi}{180}$	-0.05	0.0	0.0
7	3	4	0.2	-0.05	0.2	0.0	0.0	-0.05	0.0	0.0
8	4	3	0.2	-0.05	0.2	0.0	0.0	-0.05	0.0	0.0
9	2	5	4.0	-0.5	14.0	-10.0	$-\frac{45\pi}{180}$	-7.0	5.0	$-\frac{5\pi}{180}$
10	5	2	4.0	-0.5	14.0	-10.0	$-\frac{45\pi}{180}$	-7.0	5.0	$-\frac{5\pi}{180}$
11	2	6	4.0	-0.5	14.0	10.0	$\frac{45\pi}{180}$	-7.0	-5.0	$\frac{5\pi}{180}$
12	6	2	4.0	-0.5	14.0	10.0	$\frac{45\pi}{180}$	-7.0	-5.0	$\frac{5\pi}{180}$
13	6	5	4.0	-0.5	14.0	0.0	0.0	-7.0	5.0	$\frac{5\pi}{180}$
14	5	6	4.0	-0.5	14.0	0.0	0.0	-7.0	5.0	$\frac{5\pi}{180}$
15	5	3	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
16	3	5	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
17	6	4	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$
18	4	6	4.0	-1.0	16.0	0.0	0.0	-8.0	5.0	$\frac{5\pi}{180}$

It is relevant to point out here that the sign of the uncertainty ϵ_T relative to the maneuver duration is negative. The choice of this sign will be explained in App. B once the global problem is formulated.

3.2.2 Evaluation using the GT-Max Simulator

In general cases, it is difficult to model the uncertainties present in motion primitives. In order to have a better idea on how uncertainties affect different trim and maneuver conditions the simulation environment provided by the GT-Max was used.

The controller architecture of the GT-Max is a combination of an inner loop, which controls the attitudes and an outer loop, which controls the translational trajectory of the vehicle. The adaptive element of the controller is a neural network, which focuses on the accurate tracking of position commands. For more information on the dynamics of the GT-Max and controller architecture the reader is referred to [19]. However, before the reader is introduced to the assessment of perturbations, or uncertainties measured through the simulator, it is important to point out what those measures correspond to.

There are two sources of uncertainty that affect the response of the simulator. Those are external perturbations such as a flight tested turbulence model which acts on the UAV and sensor noise. The uncertainty data that was recorded and assessed in this section corresponds to the actual states. Notice that there is already a discrepancy, at the level of the Kalman filter, between the estimated states and the real states (where the vehicle thinks it is in comparison to where it actually is). Based on the fact that the comparison is between the commanded states and the estimated ones, the controller might think the system reached the target states even though it is not the case. Hence, a bad estimate of the state will not be compensated for by the controller. The block diagram in Fig. 5 shows where the uncertainties originate from and where they are present.

3.2.3 Extracting the Motion Primitives

A MATLAB code was written in order to generate sequences of trims and maneuvers using the library information provided in the previous section. As such, all nominal data and sequence information were imported to the simulators external maneuver generator

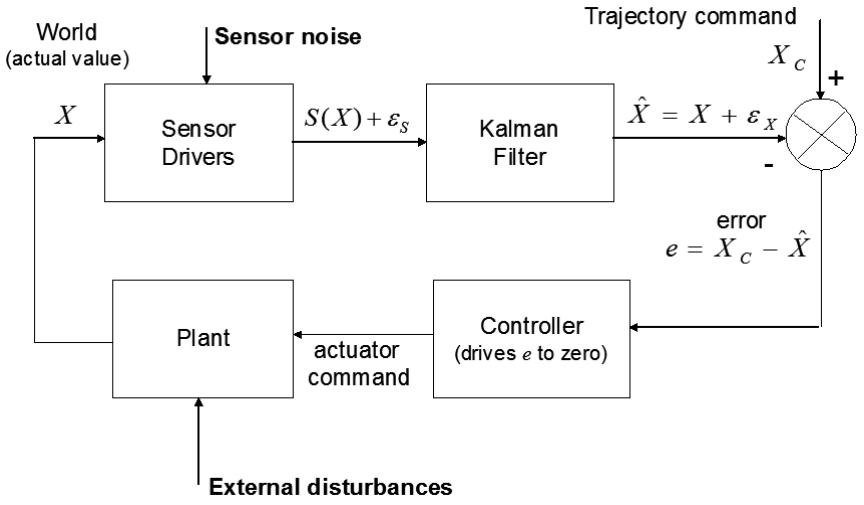


Figure 5: Origin of Uncertainties in the GT-Max

environment.

3.2.4 Trajectory generation using GT-Max

Path planning using the GT-Max happens after waypoints are specified. Smooth trajectories are generated using a kinematics based external trajectory generator. The type of waypoint chosen dictates the type of trajectory that will be generated in order to reach the waypoint.

3.2.5 Evaluating Uncertainties for Given Motion Primitive Sequence

In order to evaluate trims and maneuvers, the nominal motion primitive library described in the previous section was used. Given a sequence of trims and maneuvers, the trajectory generation begins with the initial trim. This trim is followed for a random amount of time. Once the time allocated to this trim condition is elapsed, the simulator switches to the second trim in the sequence and so on, until all the trims in the sequence are performed. The transitions between trim conditions are defined as maneuvers. All the navigation data is recorded in order to assess the actual vehicle behavior.

The trim conditions that are defined in the reference library were chosen such that attitude limits are not exceeded and rate saturation does not occur. The controller of the GT-Max is designed such that if the above mentioned holds, the system behaves like a second order linear dynamic system, whose settling time is ideally constant. Therefore, when a

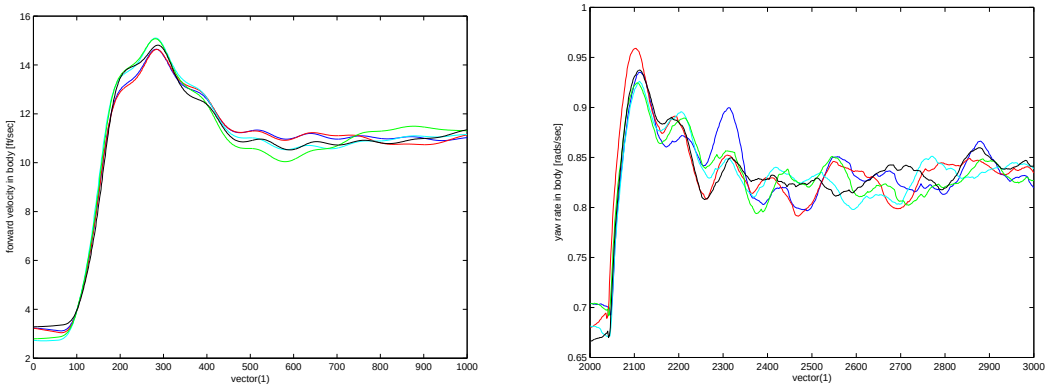
transition between two forward flights is studied, the associated maneuver duration will be equal to the settling time required for the system to reach the desired velocity. Let ΔV be the desired velocity change between the trim condition the maneuver issues from and the relative equilibria the maneuver leads to. For all ΔV, the settling time is constant as long as the velocities are below actuator saturation limits. Hence, in the absence of uncertainties, the settling time, or in other words, the maneuver duration between any forward flight in the library will be equal to the same constant. Similarly, the maneuver duration between two turning flights at different angular velocities will remain constant for any turning flight in the library given that the actuator limits are not exceeded. The associated maneuver duration errors are studied in upcoming sections.

3.2.6 Populating the Library

Per definition, maneuvers are transitions between one trim to another. Within the data collected, maneuvers were recognized by a sudden change in velocities if the maneuver was either acceleration or deceleration , or by a variation of the turn rates for banking maneuvers.

Several maneuvers were recorded and identified during the flight tests in the simulated environment. It was noticed that each maneuver type is foreseeable in terms of the displacement and duration and is repeatable as can be seen in Fig. 7.

The uncertainties on motion primitives were computed as maximum deviations. Those



(a) Forward Velocity (ft/s) (b) Yaw Rate (rads/s)

Figure 6: Maneuvers are repeatable

include, as mentioned earlier, uncertainties on the heading change, finite duration and finite displacement. For trims, the same procedure was used but the uncertainties were computed only on velocities. Hence the following uncertainties were added to the library:

- $\epsilon_{u_{fwd}}$, uncertainty on u_{fwd},
- $\epsilon_{u_{side}}$, uncertainty on u_{side},
- ϵ_r, uncertainty on r.

For maneuvers, were added:

- ϵ_T, uncertainty on T,
- $\epsilon_{\Delta \boldsymbol{x}_p}$, uncertainty on $\Delta \boldsymbol{x}_p$,
- $\epsilon_{\Delta\psi}$, uncertainty on $\Delta\psi$.

Table 9 and Table 10 give the updated version of the motion primitive library with uncertainties added.

Table 9: Trim Library obtained through simulations with the GT-Max - Experiment based

q	u_{fwd} [ft/s]	u_{side} [ft/s]	r [rad/s]	$\epsilon_{u_{\text{fwd}}}$ [ft/s]	$\epsilon_{u_{\text{side}}}$ [ft/s]	ϵ_r [rad/s]
1	0	0	0	0.128	0.189	0.006
2	3	0	0	0.620	0.619	0.027
3	10	0	0	0.610	0.529	0.020
4	3	0	-0.1	0.467	0.500	-0.017
5	3	0	0.1	0.545	0.418	0.015
6	10	0	-0.6	1.933	2.357	-0.020
7	10	0	0.6	2.462	1.631	0.014

Table 10: Maneuver Library obtained through simulations with the GT-Max - Experiment based

p	q_{from}	q_{to}	$\Delta \boldsymbol{x}_{p_x}$ [ft]	$\Delta \boldsymbol{x}_{p_y}$ [ft]	$\Delta\psi$ [rad]	$\epsilon_{\Delta \boldsymbol{x}_{p_x}}$ [ft]	$\epsilon_{\Delta \boldsymbol{x}_{p_y}}$ [ft]	$\epsilon_{\Delta\psi}$ [rad]	T [s]	ϵ_T [s]
1	2	1	6.095	0.135	0.010	-0.567	-1.939	-0.079	4.0	-0.8
2	1	2	6.255	0.067	0.007	-0.612	-1.886	-0.080	4.0	-0.8
3	1	3	19.652	0.082	0.014	-4.998	-1.221	-0.036	4.0	-1.2
4	3	1	21.087	0.034	0.022	-4.997	-1.001	-0.033	4.0	-1.2
5	3	2	25.953	0.119	0.029	-6.126	-1.612	-0.067	4.0	-1.2
6	2	3	26.170	0.102	0.019	-6.078	-1.603	-0.065	4.0	-1.2
7	4	2	8.749	-0.673	-0.134	-6.089	1.070	-0.048	3.0	-0.6
8	2	4	8.986	-0.643	-0.151	-6.116	1.338	-0.051	3.0	-0.6
9	2	5	8.856	0.671	0.144	-0.189	-1.326	0.046	3.0	-0.6
10	5	2	8.928	0.625	0.152	-0.195	-0.978	0.044	3.0	-0.6
11	4	5	8.974	-0.030	-0.013	-0.270	0.541	-0.002	3.0	-0.8
12	5	4	9.009	0.103	0.009	-0.269	-0.451	-0.002	3.0	-0.8
13	6	3	23.437	-12.233	-0.879	-1.232	9.706	-0.270	3.0	-0.9
14	3	6	24.324	-11.750	-0.900	-1.287	4.920	-0.272	3.0	-0.9
15	3	7	22.993	12.975	0.893	-1.449	-3.116	0.234	3.0	-0.9
16	7	3	24.494	11.532	0.791	-3.059	-3.157	0.229	3.0	-0.9
17	6	7	29.801	0.039	0.003	-2.892	-2.392	-0.006	3.0	-0.9
18	7	6	30.181	0.143	0.006	-4.386	-1.634	-0.006	3.0	-0.9
19	7	5	21.011	5.871	1.349	-2.525	-3.866	0.344	4.0	-1.2
20	5	7	9.340	19.857	1.400	-6.405	-1.492	0.345	4.0	-1.2
21	6	4	21.020	-5.913	1.409	-1.049	6.724	-0.323	4.0	-1.2
22	4	6	9.399	-19.709	1.377	-4.029	6.071	-0.368	4.0	-1.2

3.3 Trend Identification using the Simulator

It was pointed out that the libraries used within this investigation are limited to the choice of trim and maneuver conditions that are defined. This results in a very limited flight envelope since the MA framework only allows limited amount of actions given the size of the library. In order to add more motion primitives to the MA framework, one could investigate if there is any trend in the execution of maneuvers and the uncertainties that would allow us to compute maneuver characteristics, knowing the boundary trim conditions. For example, one could simulate a transition between two forward flights at different increments of speed and check the displacement and heading changes as well as uncertainties associated with those. This section deals with finding out if there are any trends that would allow us to predict the characteristics of a greater number of motion primitives.

3.3.1 Analysis of Trim Conditions

In order to assess what are the uncertainties associated with trim conditions, we flew the simulator at different speeds. Fig. 7 suggests that, to the difference of the assumption made in the kinematics based library, as the speed of the vehicle increases, the uncertainties, as percent errors relative to the desired velocity, decrease for the given system. Let us consider the sole case when there is a constant wind velocity from the side of the airplane. It is trivial in this case that the effect of the wind velocity on the overall deviation of the vehicle from its mean or desired forward velocity will be greater while the vehicle flies slow. A fast flying vehicle will be more stable and therefore will be less affected by the constant wind perturbing its states.

3.3.2 Analysis of Maneuver Conditions

Consider a vehicle constrained to fly straight at different speeds separated by intervals of 8 ft/s. Fig. 8 represents the required maneuver displacements as a function of the commanded forward velocity the maneuvers issue from. Furthermore, the scenario that was considered is composed of two different sections:

- the vehicle first accelerates to a fixed maximum velocity,

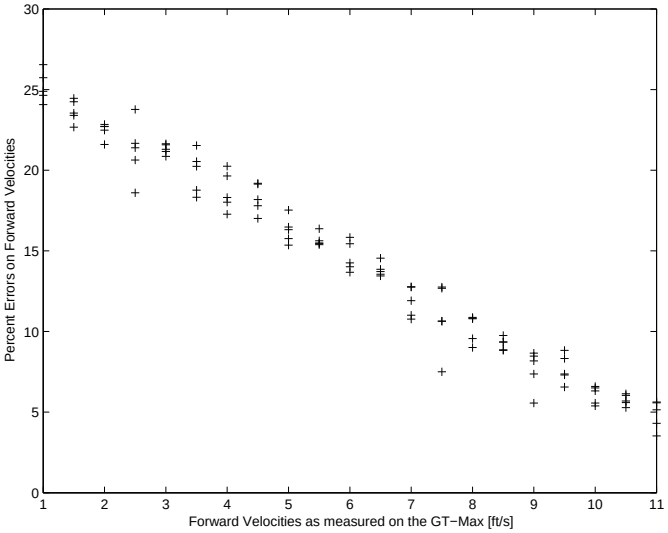


Figure 7: Percent errors in forward velocities

- the vehicle decelerates to the initial speed until it reaches the initial forward velocity.

We define ΔV in the legend, as the the velocity increment that was fixed to 8 ft/s for this example. Other scenarios with different values for ΔV were also conducted but will not be shown in this work. It can be concluded that the required displacement to reach the desired exit velocity for a given ΔV increases as the forward speed the maneuver issues from becomes larger in magnitude. It is relevant to point out that the displacements measured are relative to an inertial coordinate system, whose origin is the initial condition. The errors presented in Fig. 8 are percent errors relative to the displacements required. Their analysis exhibits two phenomena. On the one hand, at very low speeds, perturbations have a greater impact on the displacement as far as magnitude of the percent errors are concerned. This explains why the percent error increases as the speeds decrease. On the other hand, a transition between two fast forward trims is also characterized by a larger percent error on the displacement.

Now, looking to the maneuver durations, as the velocities and turn rates do not exceed actuator limits, the maneuver durations remain unchanged as explained earlier. The mean values for durations from a slow forward flight to a faster forward flight were found to be 4 seconds, the maneuver durations for turning flights at unchanged forward velocity in body axes were found to be 3 seconds. Finally, since the settling time for forward flights is greater

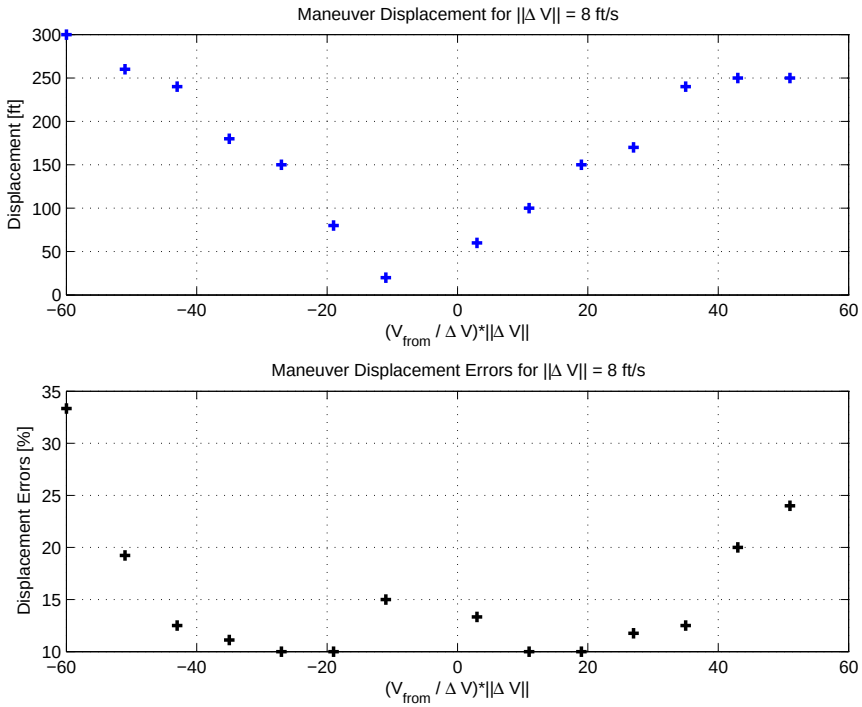


Figure 8: Maneuver displacement and associated errors

than turning flights, the mean maneuver durations associated with maneuver connecting a slow turning flight to a fast turning flight will be of 4 seconds. As far as errors associated with maneuver durations are concerned, they follow a similar pattern to displacement errors. Close to hovering, the error on the maneuver duration is about 30%. Transitioning between two fast turning flight is also about 30% in duration error. The library constructed above takes into account those numbers.

Finally we considered a vehicle flying at constant forward velocity in body axes but subject to an increasing turning rate. We will increment the turning rates by 0.1 rads/s. The resulting maneuver characteristics are presented in Fig. 9. This figure suggests that as the turning rate increases, the relative required maneuver angular displacement $\Delta\psi$ as well as the percent error on maneuver durations increase. In fact, as the yaw rate increases while the forward velocity remains constant, the radius of turn becomes smaller and smaller. As a consequence, the maneuver the vehicle has to perform in order to increase its angular rate, becomes more and more aggressive, hence the increase in both maneuver duration error and heading change.

The library constructed here through simulations provides an order of magnitude of errors

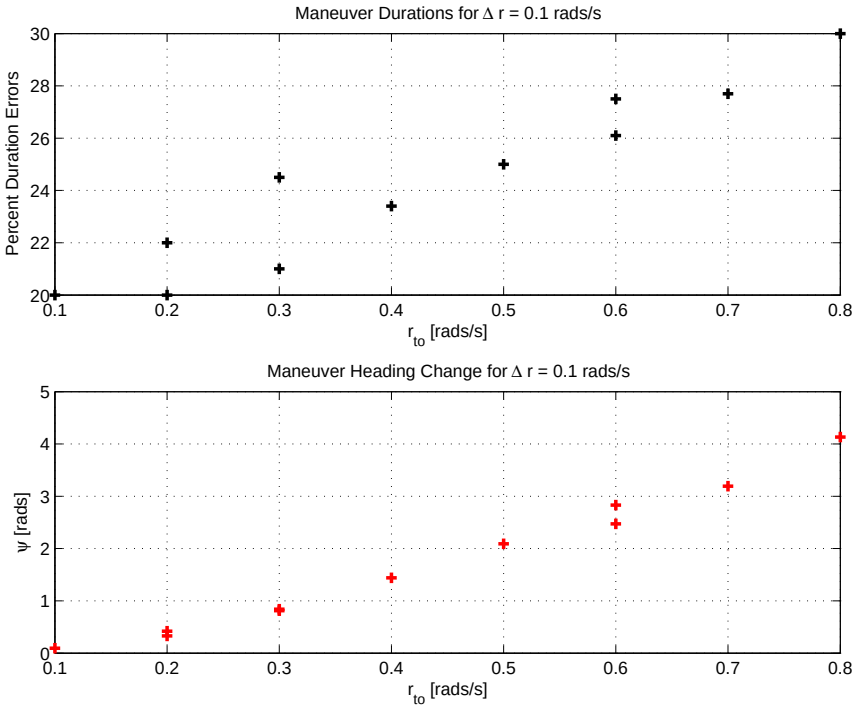


Figure 9: Maneuver heading change and duration errors for turning flights

in motion primitives as well as maneuver characteristics. Obtaining a more accurate library would require extended statistical analysis which is not the motivation of this research. However, the provided data is pertinent enough within the scope of this work and validates the developed approach.

CHAPTER IV

ROBUST OPTIMAL CONTROL IN THE PRESENCE OF UNCERTAINTIES

4.1 Optimal Control Theory

An Optimal Control Problem starts with the definition of the characteristics of the system to be controlled. This includes the dynamics of the system and the boundary conditions it is subject to. Once the system to be optimally controlled is known, the constraints it is subject to and possible alternatives must be clarified. This should be followed by the description of the task to be accomplished. At last, a clear statement of the criterion for judging optimal performance must be established [3].

4.2 Classical Optimal Control

The classical optimal control problem can be defined as follows

Problem 4.1 (Non Robust Optimal Control)**.**

$$
\begin{array}{lrll}
\min\limits_{\boldsymbol{u}} & J^{NR} = & J^{NR}(\boldsymbol{x},\boldsymbol{u}) & \textit{performance index} \\
s.t.: & \dot{\boldsymbol{x}} = & f(\boldsymbol{x},\boldsymbol{u},\bar{\boldsymbol{p}}) & \textit{system equations} \\
& \boldsymbol{x}(0) = & \boldsymbol{x}_0 & \textit{initial conditions} \\
& \boldsymbol{c}(\boldsymbol{x},\boldsymbol{u}) \in & [\boldsymbol{c}_l,\boldsymbol{c}_u] & \textit{constraints} \\
& \psi(\boldsymbol{x}(t_f)) \in & [\psi_{t_l},\psi_{t_u}] & \textit{terminal constraints} \\
& \boldsymbol{x} \in & [\boldsymbol{x}_l,\boldsymbol{x}_u] & \textit{state bounds} \\
& \boldsymbol{u} \in & [\boldsymbol{u}_l,\boldsymbol{u}_u] & \textit{control bounds}
\end{array}
$$

for some ***nominal*** *model parameters* $\bar{p}$*.*

The solution to Problem 4.1 yields the optimal control policy $\boldsymbol{u}^{NR}_{opt}$, the optimal trajectory $\boldsymbol{x}^{NR}_{opt}$ and the corresponding optimal cost $J^{NR}_{opt} = J^{NR}(\boldsymbol{x}^{NR}_{opt}, \boldsymbol{u}^{NR}_{opt})$.

In reality, the model parameters $\boldsymbol{p}$ are uncertain, i.e.

$$\boldsymbol{p} = \bar{\boldsymbol{p}} + \epsilon_{\boldsymbol{p}}$$

where $\epsilon_{\boldsymbol{p}}$ is a stochastic variable. We could also consider other forms of uncertainty, e.g. in the functional form of the system equations

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) + \epsilon_f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p})$$

however for simplicity we restrict the attention to the sole case of model parameter uncertainty in the following.

In general, applying the optimal nominal policy $\boldsymbol{u}_{opt}^{NR}$ to the non-nominal case results in a sub-optimal trajectory. In other words, solving the initial value problem

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}, \boldsymbol{u}_{opt}^{NR}, \boldsymbol{p}), \quad \boldsymbol{p} \neq \bar{\boldsymbol{p}} \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \end{aligned}$$

results in a trajectory $\boldsymbol{x}_{\epsilon}^{NR} \neq \boldsymbol{x}_{opt}^{NR}$ such that

$$J^{NR}(\boldsymbol{x}_{\epsilon}^{NR}, \boldsymbol{u}_{opt}^{NR}) > J_{opt}^{NR}$$

Furthermore, there is no guarantee that the constraints will be satisfied, i.e. in general,

$$\begin{aligned} &\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) \notin [\boldsymbol{c}_l, \boldsymbol{c}_u] \\ &\psi(\boldsymbol{x}(t_f)) \notin [\psi_{t_l}, \psi_{t_u}] \\ &\boldsymbol{x}_{\epsilon}^{NR} \notin [\boldsymbol{x}_l, \boldsymbol{x}_u] \end{aligned}$$

Consequently, the solution to Problem 4.1 is non-robust to perturbations (uncertainties) in the system parameters $\boldsymbol{p}$

4.3 Robust Optimal Control

We would like to hedge against the possible effects of uncertainties. However, we have no detailed knowledge of the uncertainties, such as statistical distributions etc..., and we do not want to assume that we will ever have this information, since it might be hard if not impossible to obtain.

We are therefore ready to trade some performance, as measured by the performance index $J^{NR}(\boldsymbol{x}, \boldsymbol{u})$, to gain in robustness, i.e. we will be satisfied with any value of the performance index J^{NR} such that

$$J^{NR}_{opt} \leq J^{NR} \leq \alpha J^{NR}_{opt}, \quad \alpha \geq 1$$

where α measures the acceptable performance loss. Clearly, $J^{NR} \approx J^{NR}_{opt}$ would be preferable to $J^{NR} \approx \alpha J^{NR}_{opt}$ but this is not strictly necessary nor required.

Under this perspective, we will consider the family of uncertain parameters $\boldsymbol{p}$ as bounded by an unknown value γ

$$(\boldsymbol{p} - \bar{\boldsymbol{p}})^2 \leq \gamma^2 \quad \gamma \geq 0$$

γ is referred to as the uncertainty level. The greater its value, the greater the uncertainty with respect to the nominal value of the system parameters $\bar{\boldsymbol{p}}$.

Robust optimal control maximizes the uncertainty level γ that still guarantees a certain acceptable performance decrease with respect to the non-robust case, i.e. it finds the maximum uncertainty level that satisfies the trade-off we are ready to accept. We can now formulate the robust optimal control problem as follows.

Problem 4.2 (Robust Optimal Control)**.**

$$\begin{aligned}
\max_{\boldsymbol{u}} \quad & J^R = \gamma && \textit{performance index} \\
s.t.: \quad & \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) && \textit{system equations} \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 && \textit{initial conditions} \\
& (\boldsymbol{p} - \bar{\boldsymbol{p}})^2 - \gamma^2 \leq 0 && \\
& \gamma \geq 0 && \\
& J^{NR}(\boldsymbol{x}, \boldsymbol{u}) \leq \alpha J^{NR}_{opt} && \textit{acceptable performance loss} \\
& \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) \in [\boldsymbol{c}_l, \boldsymbol{c}_u] && \textit{constraints} \\
& \psi(\boldsymbol{x}(t_f)) \in [\psi_{t_l}, \psi_{t_u}] && \textit{terminal constraints} \\
& \boldsymbol{x} \in [\boldsymbol{x}_l, \boldsymbol{x}_u] && \textit{state bounds} \\
& \boldsymbol{u} \in [\boldsymbol{u}_l, \boldsymbol{u}_u] && \textit{control bounds}
\end{aligned}$$

The solution to Problem 4.2 results in the robust control policy $\boldsymbol{u}_{opt}^{R}$, trajectory $\boldsymbol{x}_{opt}^{R}$ and maximum allowable uncertainty level γ_{max} with respect to the performance loss defined.
In order to verify the robustness to perturbations or uncertainties on model parameters of the two optimal policies u_{opt}^{NR} and u_{opt}^{R}, we set

$$\boldsymbol{p}_{\gamma} = \bar{\boldsymbol{p}} \pm \gamma_{max}$$

which is the maximum parameter deviation with respect to the nominal value $\bar{\boldsymbol{p}}$ that still satisfies the performance requirement.
We compute the trajectories generated by the control policies u_{opt}^{NR} and u_{opt}^{R} in the non-nominal maximum deviation case by solving the initial value problems

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}_{opt}^{NR}, \boldsymbol{p}_{\gamma_{\pm}})$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

yielding $\boldsymbol{x}_{\gamma_{\pm}}^{NR}$, and

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}_{opt}^{R}, \boldsymbol{p}_{\gamma_{\pm}})$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

that yields $\boldsymbol{x}_{\gamma_{\pm}}^{R}$.
At this point, we can compare the performance indices

$$\max(J^{NR}(\boldsymbol{x}_{\gamma_+}^{NR}, \boldsymbol{u}_{opt}^{NR}), J^{NR}(\boldsymbol{x}_{\gamma_-}^{NR}, \boldsymbol{u}_{opt}^{NR}))$$
$$\max(J^{R}(\boldsymbol{x}_{\gamma_+}^{R}, \boldsymbol{u}_{opt}^{R}), J^{R}(\boldsymbol{x}_{\gamma_-}^{R}, \boldsymbol{u}_{opt}^{R}))$$

and we can verify the constraint satisfaction or lack thereof. Therefore, the robust optimal control problem is a two step process:

1. First, one solves the nominal non-robust problem (see Problem 4.1) to identify the theoretical best possible value of the performance index J_{opt}^{NR}, which is, however, practically not achievable and non-robust, due to the presence of uncertainties.

2. Second, one trades some performance to gain in robustness, by solving the robust problem (see Problem 4.2).

The result is a control policy that hedges against uncertainties in the maximum possible way allowed by the acceptable performance loss.

4.4 Example

Problem 4.3 (Non-Robust Problem)**.** *Find*

$$\begin{aligned} \min_{x,y} \quad & J = x^2 + y^2 \\ s.t.: \quad & y = x + \bar{a} \end{aligned}$$

This problem is equivalent to finding the minimum of the curve generated by the intersection of the plane passing through $y = x + \bar{a}$ and parallel to J and the paraboloid $J = x^2 + y^2$.

Solution. *We can rewrite the augmented performance index as*

$$\hat{J} = x^2 + y^2 + \lambda(y - x - \bar{a})$$

differentiating we obtain

$$\partial\hat{J} = 2x\partial x + 2y\partial y + \partial\lambda(y - x - \bar{a}) + \lambda(\partial y - \partial x)$$

The optimal solution is found by setting

$$\begin{aligned} \partial x \rightarrow \quad & 2x - \lambda = 0 \\ \partial y \rightarrow \quad & 2y + \lambda = 0 \\ \partial\lambda \rightarrow \quad & y - x - \bar{a} = 0 \end{aligned}$$

which yields

$$\begin{aligned} x = \quad & -\frac{\bar{a}}{2} \\ y = \quad & \frac{\bar{a}}{2} \\ \lambda = \quad & -\bar{a} \\ J_{opt} = \quad & \frac{\bar{a}^2}{2} \end{aligned}$$

Problem 4.4 (Robust Problem)**.** *Find*

$$\begin{aligned} \max_{x,y} \quad & \gamma \\ s.t.: \quad & y = x + \bar{a} \\ & (a - \bar{a})^2 - \gamma^2 \leq 0 \\ & \gamma \geq 0 \\ & J = x^2 + y^2 \leq \alpha \frac{\bar{a}^2}{2} \end{aligned}$$

This problem can be interpreted as finding the maximum deviation of a that will still guarantee that the minimum of the intersection curve is not higher than the value $\alpha\frac{\bar{a}^2}{2}$

Solution. *We can rewrite the augmented performance index as*

$$\hat{J} = \gamma + \lambda(y - x - a) + \mu[(a - \bar{a})^2 - \gamma^2 + r^2] + \nu[x^2 + y^2 - \alpha\frac{\bar{a}^2}{2} + s^2]$$

The optimal solution is found by setting

$$\begin{aligned} \partial x \rightarrow \quad & 2\nu x - \lambda = 0 \\ \partial y \rightarrow \quad & 2\nu y + \lambda = 0 \\ \partial \lambda \rightarrow \quad & y - x - a = 0 \\ \partial \gamma \rightarrow \quad & 1 - 2\mu\gamma = 0 \\ \partial \mu \rightarrow \quad & (a - \bar{a})^2 - \gamma^2 + r^2 = 0 \\ \partial a \rightarrow \quad & -\lambda + 2\mu(a - \bar{a}) = 0 \\ \partial \nu \rightarrow \quad & x^2 + y^2 - \alpha\frac{\bar{a}^2}{2} + s^2 = 0 \\ \partial r \rightarrow \quad & 2r\mu = 0 \\ \partial s \rightarrow \quad & 2s\nu = 0 \end{aligned}$$

which yields

$$\begin{aligned} x = & \quad -\frac{\bar{a}\sqrt{\alpha}}{2} \\ y = & \quad \frac{\bar{a}\sqrt{\alpha}}{2} \\ a = & \quad \pm\bar{a}\sqrt{\alpha} \\ J = & \quad \alpha\frac{\bar{a}^2}{2} \end{aligned}$$

Notice that, $\bar{J} = J_{opt} = \frac{\bar{a}}{2}$ which yields $J = \alpha\bar{J}$. Moreover, $\bar{J} = \bar{r}^2$. Combining both, we obtain $r = \bar{r}\sqrt{\alpha}$ ($\bar{(\cdot)}$ terms denote nominal solutions to Problem 4.3).

4.5 Robust Optimal Control for the Maneuver Automaton

For the maneuver automaton representation of the dynamics of the system, the uncertainty is present in motion primitive characteristics as defined in Section 2.3. In assessing the effect of uncertainties in order to add robustness, one can choose amongst two different ways to look at the problem:

1. Either $\gamma^{(i)}$ is the uncertainty associated with each motion primitive parameter. For a given sequence, this will result in a vector of uncertainties, $\mathbf{\Gamma} = [\gamma^1, \ldots, \gamma^{n_T+n_M}]^T$.

2. Either γ is a scaling factor which multiplies the variation in motion primitive parameters.

For the scope of this research we have chosen to use a single scaling factor that will be used for each motion primitive parameter. The motivation under this choice is the fact that it results in a more computationally feasible problem. The justification to this assumption will be provided in App. B, once the motion planning problem for the MA is provided.

CHAPTER V

MOTION PLANNING

5.1 Mechanical Control Systems and Motion Planning

For underactuated systems, the trajectory planning problem can be formulated as a nonlinear optimization problem, where the control history parameters are coefficients of certain basis functions and the state trajectory is obtained by integrating the dynamic equations of motion, which are expressed as ODEs. Alternatively the state history can be parametrized by coefficients of various basis functions with second-order constraints imposed on the state and control variables (for extended information see [4]) . Heuristic randomized searches [18, 20] give up on optimality and completeness to achieve a reduction in average computational time. Frazzoli et al. [16, 13, 14] search a graph of motion primitives, which is the MA framework, to find motion plans for time-invariant dynamical systems. Other approaches to trajectory planning involve using special structure in the equations of motion, such as chained-form or differential flatness [11], to simplify the trajectory planning problem.

5.2 General Formulation

Let us consider a pair (A, B) of admissible configurations. A motion planning program must take as input (A, B) and should produce as output, a description of a motion of the system which starts at configuration A and ends at configuration B. Hence, the motion of the system is a function of the input pair (A, B).

5.3 Motion Planning for UAV's

For UAVs, the system in question is a time invariant mechanical system that we denote S. Let us also define the state space of S as X. The behavior, over time, of a UAV can be described as a set of differential equations on the states $\boldsymbol{x} \in X$, such that, at any instant

in time, the states satisfy

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$$

where $\boldsymbol{u}$ denotes control inputs to the system. For the general case, the motion planning will be performed from some initial position in space that we term as $\boldsymbol{x}_i$ to some final target position $\boldsymbol{x}_f$ given the control input $\boldsymbol{u}$. Moreover,

$$\begin{aligned} \boldsymbol{x}_i &= \boldsymbol{x}(t_0), & \boldsymbol{x}_i \in X \qquad & t_0 \in \mathbb{R}^+ \\ \boldsymbol{x}_f &= \boldsymbol{x}(t_f), & \boldsymbol{x}_f \in X \qquad & t_f \in \mathbb{R}^+ \end{aligned}$$

Generally, the system will be subjected to constraints, equality or inequality, on its states and controls. These constraints are expressed as

$$\begin{aligned} \boldsymbol{c}_{eq}(\boldsymbol{x}, \boldsymbol{u}) &= 0 \\ \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) &\leq 0 \end{aligned}$$

We would like to plan the trajectory such that the vehicle not only meets the constraints but also meets them while minimizing some cost. As such the performance of the generated trajectory takes importance. Thus, the solution to the problem has to not only be feasible but also minimize the cost. In the scope of this research, the cost is defined as the time it takes to perform the mission. Consequently, the problem becomes an optimal control problem where it is desired to minimize the time performance of the vehicle given that the vehicle meets the boundary conditions and constraints [5]. Generically, let us define the performance index as $J(\boldsymbol{x}, \boldsymbol{u})$. The formulation of the optimal motion planning problem in a free environment is given next.

Problem 5.1. *Given a mechanical control system S, an input position pair $\boldsymbol{x}_i, \boldsymbol{x}_f \in X$, find an input control $\boldsymbol{u}(t)$ such that, $\forall\, t \in \mathbb{R}^+$*

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{2a}$$

$$\boldsymbol{c}_{eq}(\boldsymbol{x}, \boldsymbol{u}) = 0 \text{ or } \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) \leq 0 \tag{2b}$$

$$J(\boldsymbol{x}, \boldsymbol{u}) \text{ is minimized} \tag{2c}$$

In summary, the motion planning and guidance system of an autonomous vehicle enables the vehicle to build and execute a motion plan.

In the scope of this research, using the MA vocabulary, we define the motion planning problem as finding the sequence of primitives, stored within the library, that will enable us to minimize some cost.

5.4 Motion Planning Using the Maneuver Automaton

The MA modeling of the dynamics of the vehicle was chosen as its usage results in a reduction in computational complexity of the motion planning problem because there is no need for a state, control or time discretization.

For a given problem, the maximum trajectory depth, i.e. maximum number of trajectory trims will be fixed.

Definition 5.1. *We define the vector of trajectory trims as*

$$\boldsymbol{q} = \{q_1, ..., q_{n_T}\}^T$$

$$card(\boldsymbol{q}) = n_T$$

where n_T is the maximum number of trajectory trims. The vector of variable coasting times associated with each trajectory trim in $\boldsymbol{q}$ is noted as

$$\boldsymbol{\tau} = \{\tau_1, ..., \tau_{n_T}\}^T$$

$$card\,(\boldsymbol{\tau}) = n_T$$

Similarly,

Definition 5.2. *We define the vector of trajectory maneuvers as*

$$\boldsymbol{p} = \{p_1, ..., p_{n_M}\}^T$$

$$card(\boldsymbol{p}) = n_M$$

where $n_M = n_T - 1$. For all trajectory maneuvers we define the vector of fixed times as

$$\boldsymbol{T} = \{T_1, ..., T_{n_M}\}^T$$

$$card\,(\boldsymbol{T}) = n_M$$

A sequence of trims and maneuvers of fixed depth $n_T + n_M$ can then be written as:

$$\forall i \in [1, n_T + n_M] \qquad (u_n)_{n=1}^{n_T+n_M} : \begin{cases} u_i = q \in \boldsymbol{q} & \text{if} \quad i = 2n_T - 1 \quad ; \\ u_i = p \in \boldsymbol{p} & \text{if} \quad i = 2n_T \qquad . \end{cases}$$

It is important to point out that, given the above formulation, each defined sequence of motion primitives will start with a trim from the library and will end in one trim.

As $\boldsymbol{p} = \boldsymbol{p}\,(\boldsymbol{q})$, the set of all possible sequences can be expressed as a function of trims only. In other words, picking a particular sequence of possible trims determines the particular maneuvers to be used in transitioning from one to the next trim. We will call the set of all possible sequences of motion primitives as

$$\mathscr{U}_T = \{\cup\, \boldsymbol{q}_j,\, \max\,[\,\mathrm{card}(\boldsymbol{q}_j)\,] = n_T\}$$

where the index $j = 1, ..., n_{max}$ refers to one sequence of motion primitives picked within $\mathscr{U}_T$ whose $\mathrm{card}\,(\mathscr{U}_T) = n_{max}$. Notice that $\mathscr{U}_T$ is finite dimensional given that it contains a maximum number of sequences and given that each sequence has a finite number of trims and maneuvers, and hence a finite depth $n_T + n_M$. Since each sequence of motion primitives can be determined knowing the sequence of trims, we will refer, in this analysis, to the depth of the sequence, in terms of the depth of the vector of trajectory trims.

5.4.1 Dynamics within the Maneuver Automaton Framework

Omitting uncertainties, the continuous state $\boldsymbol{x}$ evolves over time as follows

$$\boldsymbol{x}_{\,2k} = \phi(\boldsymbol{w}_{\,q_k}\tau_k)\,\boldsymbol{x}_{\,2k-1} \qquad \text{in a trim condition} \qquad k = 1, ..., n_T\,; \tag{3a}$$

$$\boldsymbol{x}_{\,2k+1} = \boldsymbol{x}_{\,2k} + \Delta\boldsymbol{x}_{\,p} \qquad \text{in a maneuver} \qquad k = 1, ..., n_M\,. \tag{3b}$$

In the above expression, $\boldsymbol{w}_{q_k}$ represent the generalized velocities specific to trim q_k, $\phi\,(\,\cdot\,)$ is the exponential function that describes the dynamics within a trim condition. Adding the uncertainties, the dynamics of the system will take the form,

$$\boldsymbol{x}_{\,2k} = \phi((\boldsymbol{w}_{\,q_k} + \epsilon_{\boldsymbol{w}_{\,q_k}})\tau_k)\,\boldsymbol{x}_{\,2k-1} \qquad \text{in a trim condition} \qquad k = 1, ..., n_T\,; \tag{4a}$$

$$\boldsymbol{x}_{\,2k+1} = \boldsymbol{x}_{\,2k} + \Delta\boldsymbol{x}_{\,p} + \epsilon_{\Delta\boldsymbol{x}_{\,p}} \qquad \text{in a maneuver} \qquad k = 1, ..., n_M\,. \tag{4b}$$

where $\epsilon_{\boldsymbol{w}_{\,q_k}}$ are the uncertainties on the generalized velocities.

5.5 Optimal Control Problem

We will introduce the optimal control problem as finding the best sequence of trims that would result in reaching a target position minimizing a performance index [3]. For this problem, we are interested in finding the best sequence that would minimize the time it takes to reach the final boundary condition. To this end

Definition 5.3. *We define the cost function as*

$$J : \{\boldsymbol{q}, \boldsymbol{\tau}\} \in \{\mathscr{U}_T{}^{n_T}, \mathbb{R}^{+^{n_T}}\} \rightarrow J(\boldsymbol{q}, \boldsymbol{\tau}) \in \mathbb{R}$$

where $\boldsymbol{q} \in \mathscr{U}_T$ is a trajectory trim sequence and $\boldsymbol{\tau}$ is the vector of coasting times associated with this sequence.

Each element of $\boldsymbol{\tau}$ is associated with the corresponding trim. We will express the total time it takes to reach $\boldsymbol{x}_f$ as:

$$\sum_{i=1}^{n_T} \tau_i + \sum_{i=1}^{n_M} T_i \tag{5}$$

The first term in this expression corresponds to the total time spent in trajectory trims, which is variable, and the second term represents the total time spent in trajectory maneuvers.

CHAPTER VI

TRAJECTORY OPTIMIZATION WITH THE MANEUVER AUTOMATON

6.1 Nominal Trajectory Optimization

The cost function for the nominal case is nothing else than the total time it takes to reach the final position. Hence the optimal policy is the one that minimizes this cost whose expression is:

$$J(\boldsymbol{q}, \boldsymbol{\tau}) = \sum_{i=1}^{n_T} \tau_i + \sum_{i=1}^{n_M} T_i \tag{6}$$

The nominal optimal control problem will be formulated as:

Problem 6.1. *Given the maneuver automaton in consideration, an input position pair* $\boldsymbol{x}_i, \boldsymbol{x}_f \in X$*, find a hybrid control input pair* $\{\boldsymbol{q}, \boldsymbol{\tau}\}$ *such that,*

$$\boldsymbol{x}_{2k} = \phi(\boldsymbol{w}_{q_k} \tau_k)\, \boldsymbol{x}_{2k-1} \qquad \textit{in a trim condition} \qquad k = 1, ..., n_T\,; \tag{7a}$$

$$\boldsymbol{x}_{2k+1} = \boldsymbol{x}_{2k} + \Delta \boldsymbol{x}_p \qquad \textit{in a maneuver} \qquad k = 1, ..., n_M\,. \tag{7b}$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_i\,; \tag{7c}$$

$$\boldsymbol{x}_{2n_T} = \boldsymbol{x}_f\,. \tag{7d}$$

$$\min_{\boldsymbol{q} \in \mathscr{Q}_T} \; \min_{\boldsymbol{\tau}} J(\boldsymbol{q}, \boldsymbol{\tau}) \tag{7e}$$

The solution to Pb. 6.1 yields the optimal sequence of trim conditions $\boldsymbol{q}_{opt}$, the optimal coasting times $\boldsymbol{\tau}_{opt}$ and the corresponding optimal cost J_{opt}.

With the library of motion primitives obtained through simulations on the GT-Max, if the maximum number of trajectory trims is set to be 4, the resulting variable depth set of possible trajectory trim sequences will have 333 elements. This is the result one obtains without fixing the initial trim state. However, the number of possible sequences is further reduced if one specifies the initial trim state, with which the optimization is restricted to

start. According to this, for the problem in hand, Table 11 shows the number of trajectory primitive sequences associated with each trim in the library, for a fixed maximum trim sequence depth of 4 trims.

Table 11: Trajectory Sequences given Starting Trim

Starting Trim q	Number of trajectory sequences associated
1	35
2	57
3	57
4	46
5	46
6	46
7	46

6.2 Robust Motion Planning and Accounting for Uncertainties

To mitigate the effects of uncertainties, it is necessary to add them when implementing the optimization. As such, a new problem can be defined, where the cost to be minimized is a trade-off between the cost of the action and the uncertainties present in the cost. Hence, it is a trade of performance in order to gain in robustness.

6.2.1 Worst Case Scenario Trajectory Optimization

For this approach, the uncertainty in the maneuver duration was added to the expression of the nominal cost function. The robust cost can then be defined as

$$J^{R}(\boldsymbol{q}, \boldsymbol{\tau}, \gamma) = \sum_{i=1}^{n_T} \tau_i + \sum_{i=1}^{n_M} (T_i + \gamma \epsilon_{T_i})$$

Hence, the robust optimal control problem will be formulated as:

Problem 6.2. *Given the maneuver automaton in consideration, an input position pair*

$\boldsymbol{x}_i, \boldsymbol{x}_f \in X$, *find a hybrid control input pair* $\{\boldsymbol{q}, \boldsymbol{\tau}\}$ *such that,*

$$\boldsymbol{x}_{2k} = \phi((\boldsymbol{w}_{q_k} + \gamma\epsilon_{\boldsymbol{w}_{q_k}})\tau_k)\,\boldsymbol{x}_{2k-1} \qquad \textit{in a trim condition} \qquad k = 1, ..., n_T \tag{8a}$$

$$\boldsymbol{x}_{2k+1} = \boldsymbol{x}_{2k} + \Delta\boldsymbol{x}_p + \gamma\epsilon_{\Delta\boldsymbol{x}_p} \qquad \textit{in a maneuver} \qquad k = 1, ..., n_M \tag{8b}$$

$$\|\boldsymbol{x}_{2n_T} - \boldsymbol{x}_f\| \leq \delta \;\; \textit{where } \delta \textit{ is a tolerance} \quad \textit{on the final position error} \tag{8c}$$

$$\min_{\boldsymbol{q}\in\mathscr{U}_T} \max_{\gamma} \min_{\boldsymbol{\tau}} J^R(\boldsymbol{q}, \boldsymbol{\tau}, \gamma) \qquad \textit{where } \gamma = 0\,, -\gamma_a\,, \gamma_a \tag{8d}$$

$$\gamma_a \in [0, 1] \tag{8e}$$

The solution to Pb. 6.2 yields the optimal robust sequence of trim conditions $\boldsymbol{q}^R_{opt}$, the optimal robust coasting times $\boldsymbol{\tau}^R_{opt}$ and the corresponding optimal robust cost J^R_{opt}.
As seen in Pb. 6.2, for $\gamma_a = 1$ which corresponds to the maximum uncertainty level, γ is restrained to take the values $0, -1, 1$. When $\gamma = -1$ the uncertainties are subtracted from the motion primitive parameters, similarly, when $\gamma = 1$ all uncertainties are added and $\gamma = 0$ corresponds to the nominal case. For a given sequence of trim trajectories from the set of possible sequences, three cost functions are consecutively computed in an imbedded loop for each value of γ. Let

- J^R_{minus} be the cost associated with $\gamma = -1$,
- J^R_{plus} be the cost associated with $\gamma = 1$,
- J be the cost associated with the nominal case.

If the problem becomes infeasible for any value of γ, the sequence is considered infeasible. However, if for all three values of γ the problem is feasible for a given sequence, all three cost functions are compared. A worst case scenario formulation is used for the comparison. The optimal cost function, for the sequence in consideration, is the one that satisfies,

$$\max_{\gamma}\{J^R_{minus}, J^R_{plus}, J\}$$

therefore this optimization is referred to as worst case approach. A similar approach was investigated in [24], however an expected value formulation of the cost function was used. For the entire optimal control problem, the optimal robust sequence $\boldsymbol{q}^R_{opt}$ is the sequence associated to the minimum of the optimal worst cost scenarios.

6.3 Receding Horizon Control

In open-loop guidance, the sequence of motion primitives that will lead to the target position will be generated beforehand. As uncertainties in the motion primitives are added, the likelihood of the open-loop trajectory to reach the target is significantly reduced when the motion plan is executed. Nevertheless, one can improve the behavior of the plan using a closed-loop formulation. To this end, we will introduce a closed-loop policy using receding horizon control. For the general case, let the current state at time t be denoted as $\boldsymbol{x}(t)$. For continuous time, receding horizon optimization is summarized as follows:

1. solve an optimal control problem over a fixed future interval defined as $[t\,;\,t+\Delta t_c]$ taking into account the current and future constraints ,
2. in the resulting optimal control sequence, apply only until time $t+\Delta t_c$,
3. Determine the state reached at time $t+\Delta t_c$,
4. Repeat 1 - 2 over the future interval $[t+\Delta t_c\,;\,t+2\Delta t_c]$, starting from the advanced state $x\,(t+\Delta t_c)$.

In this definition, Δt_c refers to the time interval between two consecutive controller activations.

To the difference of receding horizon as defined, there is no horizon to track in this problem, rather a target destination to reach. We will therefore use a slightly different approach, where the time window is not of constant length but rather a variable. For any optimization performed from a different initial point, the time window is defined as $[\,T_f-t\,]$. Fig. 10 presents the receding horizon trajectory planner used for the problem in hand. It can be seen that the trajectories predicted in closed-loop all lead to the final position. Within receding horizon control, the controller is stopped, using the previous notation, after Δt_c seconds. The state reached then becomes the initial state for the optimization problem. However, there are differences given the nature of the primitives the vehicle is performing. This is due to the fact that the dynamics and therefore the evolution of the system is known

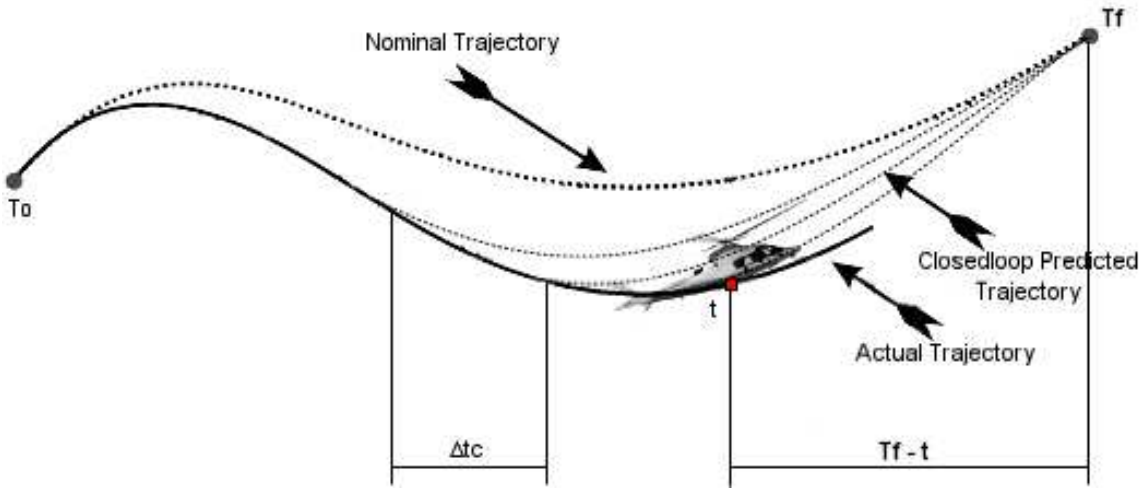


Figure 10: Receding Horizon Closed-loop control

while the vehicle is within a trim state, yet, the only information available while transitioning is fixed displacement and fixed duration. Hence, there will be two different ways the problem in hand will be treated,

- while in a trim condition, the equations of motion will directly be integrated and the next state computed,
- while the vehicle is maneuvering, the next step will not be initialized until the maneuver is completely executed.

6.3.1 Setting the tolerance in the Final Position

Even in closed-loop formulation, there is in general no guarantee that the final target position will be reached in presence of uncertainties. In fact, the constraints are initially treated as hard constraints on the final state. In closed-loop, those hard constraints are at the source of a feasibility problem. In order to improve feasibility in the solution, it is a standard technique in receding horizon to replace hard constraints by inequality constraints. Therefore, it is desired to reach the final target within a tolerance margin. When the motion planning problem is launched, the distance between the target and the initial position can be significantly big. The allowable error margin in reaching the target position will therefore be proportional to this distance. The closer the vehicle gets to the sought target

position, the smaller this allowable error will become. In other words, the tolerance on the final position becomes more conservative as we approach the target. We will estimate that the target position is reached if it is within the acceptable tolerance limits.

6.3.2 Stopping within a Trim State

The equations of motion are known and therefore can be integrated while the vehicle is within a trim state. If, for any predicted closed-loop trajectory, the vehicle ends coasting in a trim, the position for the next iteration can be computed analytically using the equations of motion (3a) depicted in Section 5.4.1.

The optimal motion primitive sequence computed in open-loop has been found with no knowledge of the uncertainties for the non-robust approach. It is in closed-loop control that the analysis of the effect of the uncertainties on the nominal trajectory will be performed. In closed-loop, adding the uncertainties to the equations of motion for a trim condition will lead the trajectory to be deviated from the nominal one. With the addition of uncertainties to the trims, the equations of motion to be used are (4a) given in Section 5.4.1

6.3.3 Stopping within a Maneuver

As a consequence of the hybrid automaton setup, once a maneuver is initiated, it has to be executed until the end. Therefore, in closed-loop formulation, the evolution of the state, once a maneuver has been initiated, is given by equation (3b) omitting uncertainties and (4b) accounting for uncertainties. If the uncertainties in motion primitives are significantly high, both closed-loop and open-loop formulations will be affected, this due to the fact that maneuvers are characterized by fixed time and displacement. Thus, any perturbation in the maneuvers will result in the vehicle drifting away from the planned, or foreseen trajectory. However, even though a closed-loop policy will correct for these deviations from the planned trajectory, it will have to wait until the maneuver is executed to do so. This is one of the disadvantages of using a MA representation of the systems dynamics, i.e. it results in only sub-optimal trajectories. Yet, the performance of the closed-loop optimization can be improved. In fact, by interpolation it is possible to predict where the vehicle is even while it is maneuvering.

CHAPTER VII

COMPARISON AND RESULTS FOR BOTH MOTION PLANNING PROBLEMS

We applied the methodology for worst case robust motion planning defined in the previous section to the two MA frameworks as presented in Chapter 2 and Chapter 3. In both cases, the vehicle will be constrained to move in a plane, from a given initial position and an initial orientation that we denote, respectively, as $\boldsymbol{x}_i$ and ψ_i to a fixed final target denoted as $\boldsymbol{x}_f$. The final orientation is not constrained.

The initial tolerance on the final position error δ is set to 3 ft, nevertheless, it is readjusted to become smaller as we approach the target position.

7.1 Results obtained with the first reference library

For more accuracy, different scenarios were tested with this library. Results for two different initial and final conditions are presented next. The MA representation that was used was presented in Chapter 3 in Table 9 for trims and Table 10 for maneuvers.

7.1.1 Results obtained with the first mission

The vehicle is constrained to move in a plane, from the given initial position $\boldsymbol{x}_i = [\,0\,\text{ft}\,;\,0\,\text{ft}\,]$ to the fixed final target $\boldsymbol{x}_f = [\,30\,\text{ft}\,;\,0\,\text{ft}\,]$. Also, the vehicle is restricted, initially, to have a heading of $\frac{3\pi}{4}$ rads such that the optimizer does not pick a straight flight as the first optimal sequence. Furthermore, the vehicle is initially constrained to fly forward at 1.0 ft/sec. There are no limitations on the final trim to be used.

7.1.1.1 Open-loop Solutions

Initially, a first open-loop solution for the optimal control problem is computed for both non-robust and the worst case scenario formulations. Those solutions correspond to the nominal or reference trajectories for our purposes. As the scaling factor γ increases, it is expected

to see that the nominal open-loop solution for the worst case optimization problem will be different from the non-robust nominal solution, as it accounts for uncertainties present in primitives. However, for this first mission, one can see that, the open-loop (nominal) optimal trajectory that was picked by the non-robust and the worst case scenario optimizations are identical. This means that the nominal trajectory primitive sequence picked by the non-robust optimization and the worst case policy is the one that is less perturbed by the uncertainties present in the motion primitives. However, as we recompute the optimal control problem in closed-loop formulation, the solutions picked by both optimizations will change as the worst case formulation will pick solutions that are more cautious.

7.1.1.2 Optimization Results

The results obtained are indicated in Fig. 11 through Fig. 15 for the non robust optimization and Fig. 16 through Fig. 20 for the worst case scenario policy. In both figures, the straight dotted lines represent maneuvers and the solid curves trim states. In Figs. 11 - 15, the green trajectory corresponds to the nominal solution to the optimal control problem and the blue trajectory represents the closed-loop solution. Similarly, in Figs. 16 - 20, the red trajectory represents the nominal solution to the worst case formulation while the green trajectory is the closed-loop solution.

7.1.1.3 Receding Horizon Control

The behavior of the vehicle, in closed-loop, is indicated in Fig. 11 through Fig. 15 and Fig. 16 through Fig. 20 for increasing γ. The examination of the closed-loop trajectories suggests that both non-robust and worst case scenario approaches pick the same solutions up to a certain point, after which they deviate. This divergence affects mainly the non-robust trajectory because the solutions are picked without knowledge of the effect and magnitude of uncertainties until the closed-loop optimization comes into play. As a consequence, the closed-loop solution to the nominal problem is highly perturbed.

The figures further indicate that the tracking performance of the worst case scenario approach is far better than the nominal one, since the primitives in the optimal sequence picked are more cautious (less affected by uncertainties) and hence, trajectories produced

are less perturbed. Therefore, the closed-loop trajectory for the worst case policy resembles the open-loop path more, while the nominal policy becomes further and further deviated. In fact, for $\gamma > 0.3$, the non robust optimization problem had to be stopped failure to have reached the target position. In comparison, the worst case policy reaches the final position even for large values of γ.

Numerical results are presented in Tables 12 through 16 each corresponding to a different scaling factor scenario. As the value of the scaling factor γ and hence the amount of uncertainty increases, the closed-loop solutions associated with the non-robust policy are stopped as the maximum number of receding horizon activations is reached. However, the worst case scenario solutions are robust to perturbations in motion primitive characteristics and the final closed-loop costs associated with the successful accomplishment of the mission objectives are lower than the worst case nominal cost. In other words, as seen in Table 15 the reference cost associated with the worst case approach is 23.34280767 seconds while the closed-loop cost is only 19.86228881 seconds.

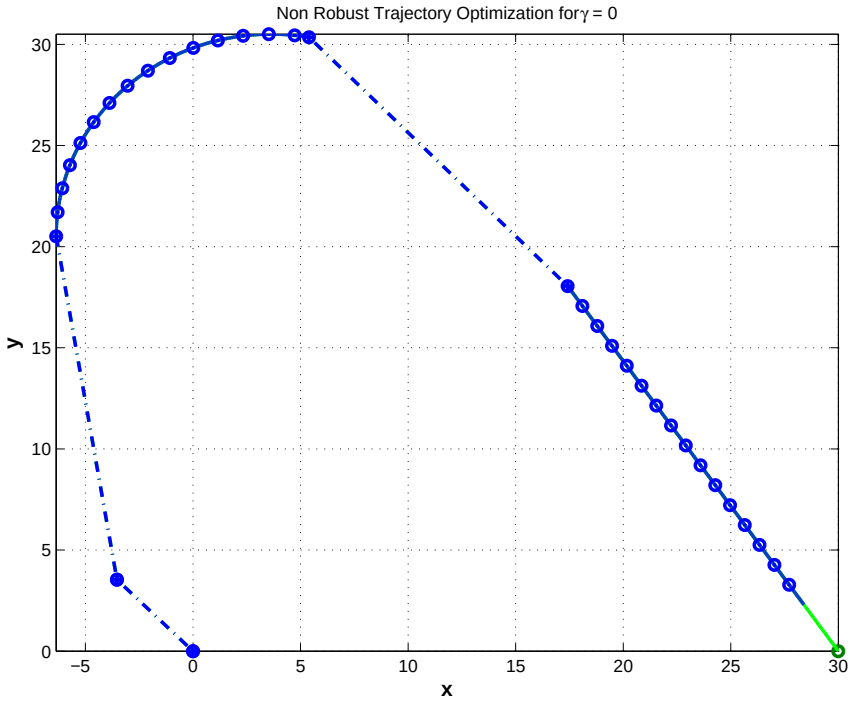


Figure 11: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.0$ - Scenario 1

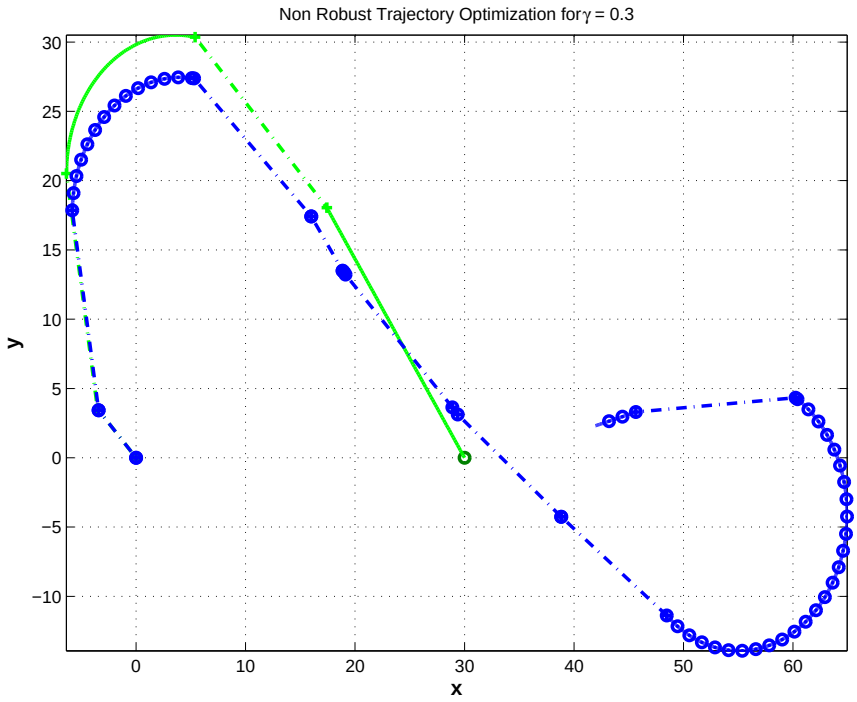


Figure 12: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.3$ - Scenario 1

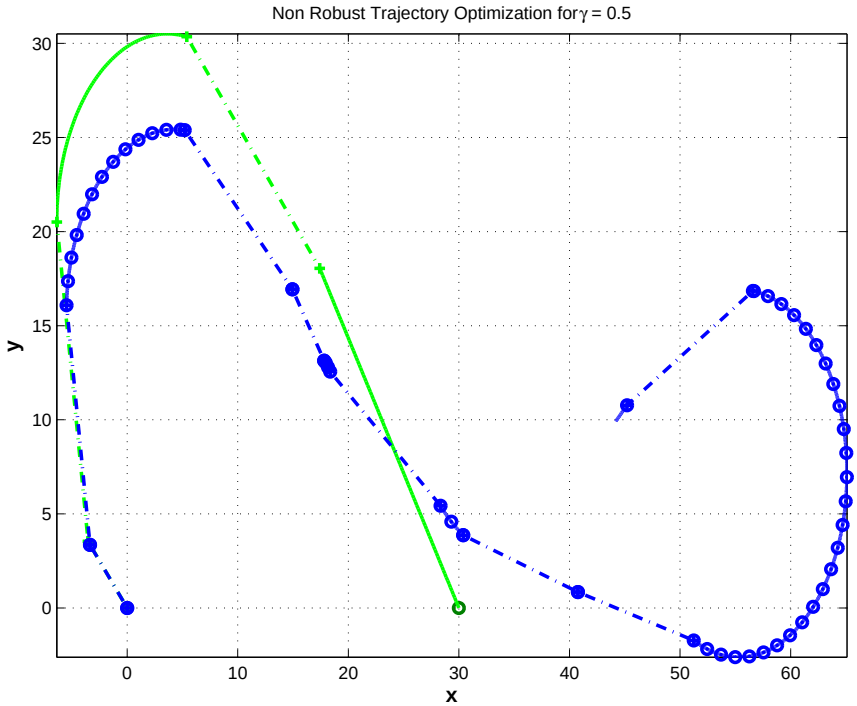


Figure 13: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.5$ - Scenario 1

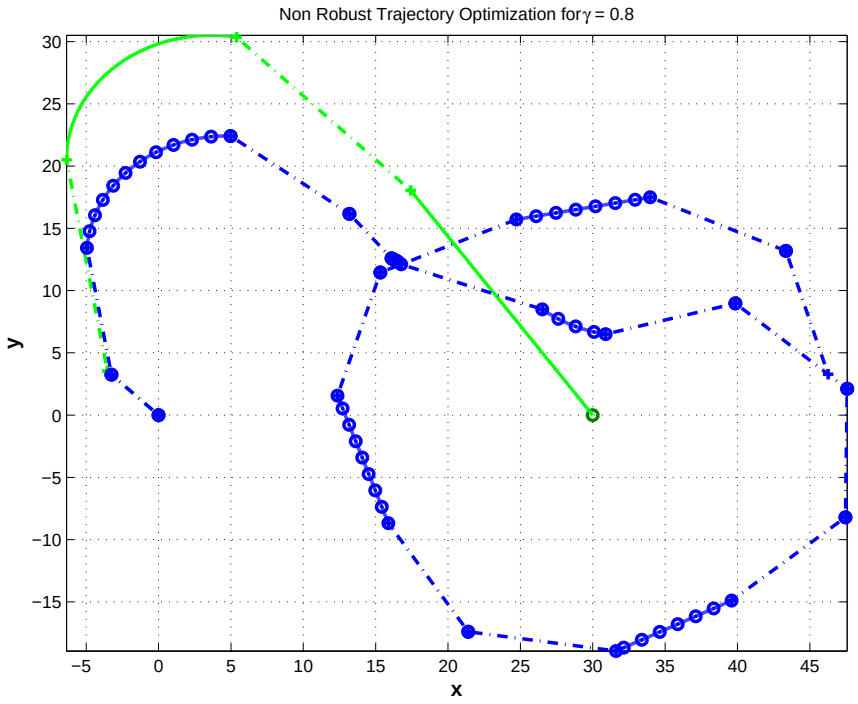


Figure 14: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.8$ - Scenario 1

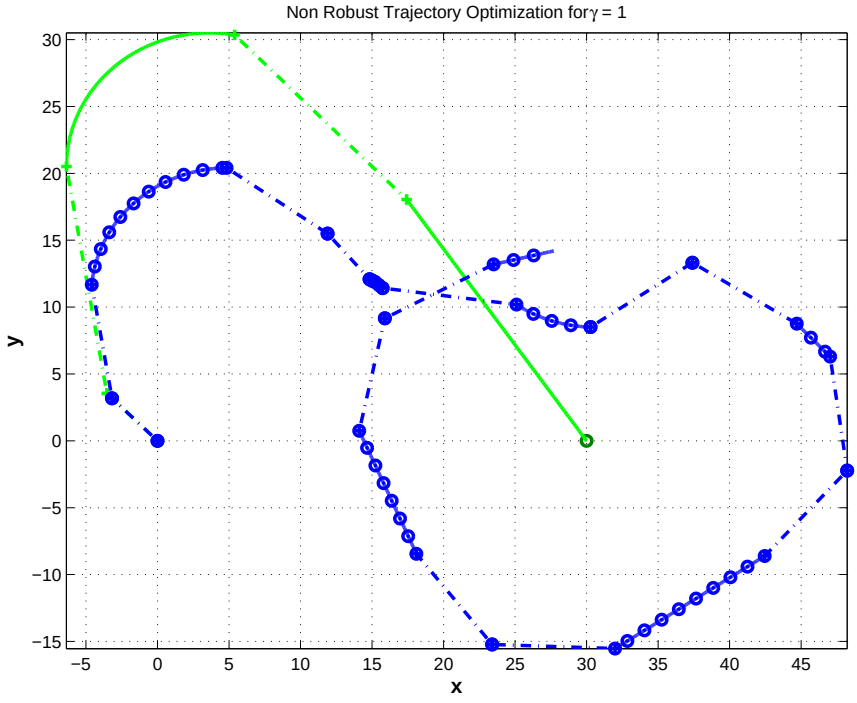


Figure 15: Closed-loop Solution to the Non Robust Approach for $\gamma = 1.0$ - Scenario 1

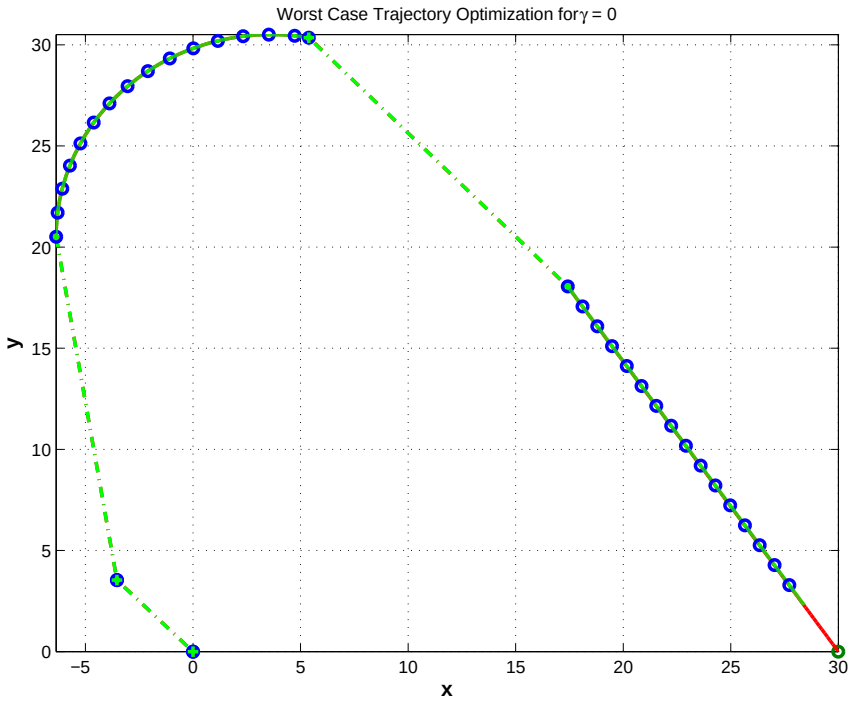


Figure 16: Closed-loop Solution to the Robust Approach for $\gamma = 0.0$ - Scenario 1

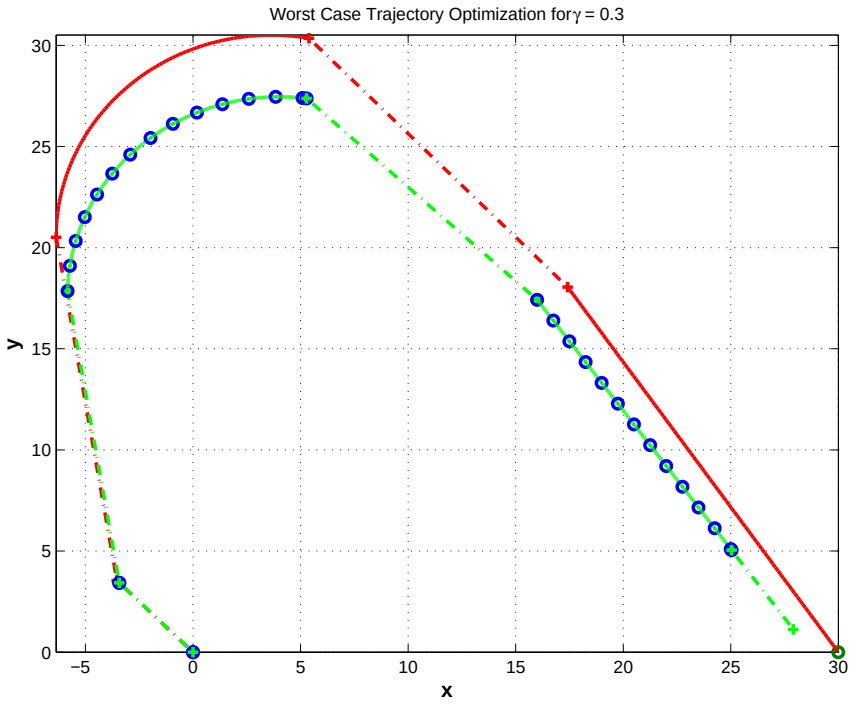


Figure 17: Closed-loop Solution to the Robust Approach for $\gamma = 0.3$ - Scenario 1

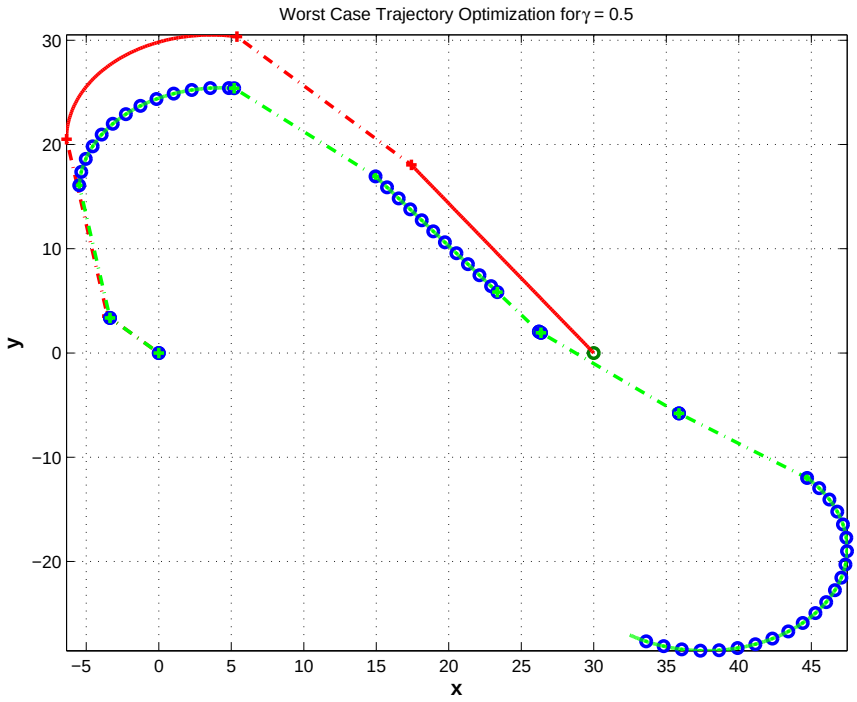


Figure 18: Closed-loop Solution to the Robust Approach for $\gamma = 0.5$ - Scenario 1

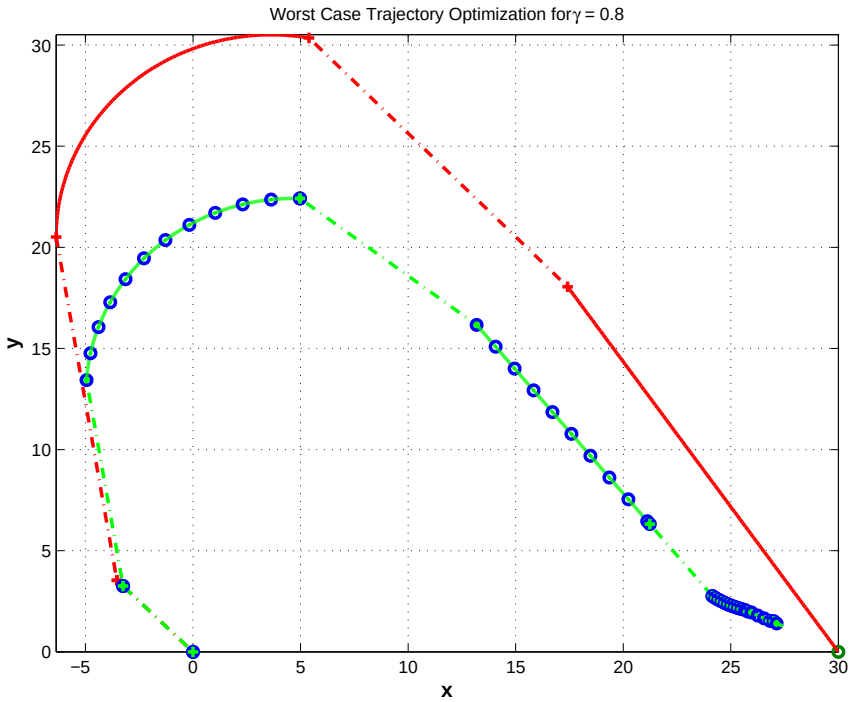


Figure 19: Closed-loop Solution to the Robust Approach for $\gamma = 0.8$ - Scenario 1

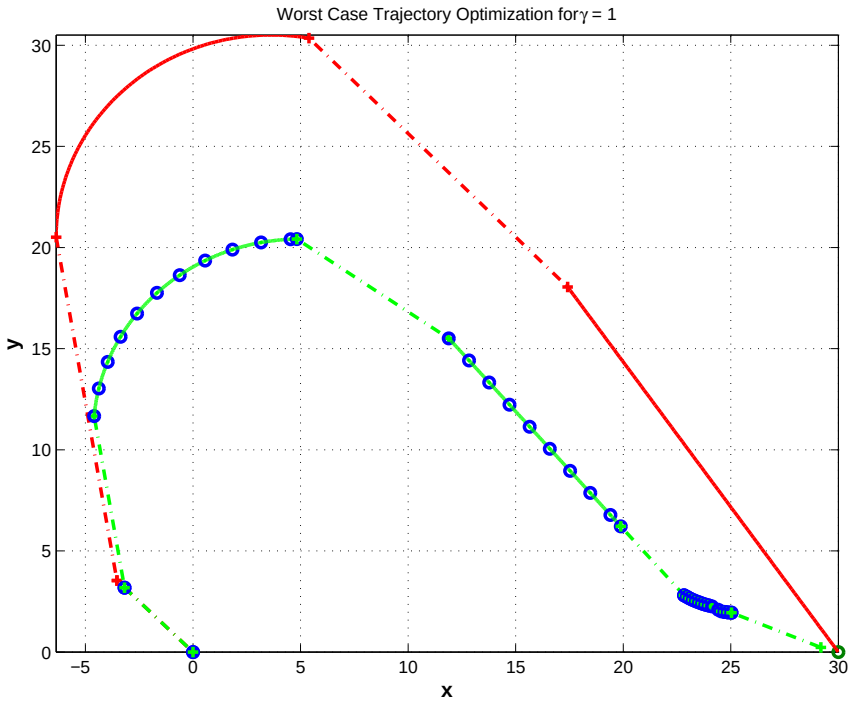


Figure 20: Closed-loop Solution to the Robust Approach for $\gamma = 1.0$ - Scenario 1

Table 12: Nominal and Closed-loop Optimal Costs for $\gamma = 0.0$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	19.16843263

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	19.16731709

Table 13: Nominal and Closed-loop Optimal Costs for $\gamma = 0.3$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	39.88946320
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
21.07064123	18.96135452

Table 14: Nominal and Closed-loop Optimal Costs for $\gamma = 0.5$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	38.93258630
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
21.93479312	31.87326767
Maximum Number of Iteration Reached	

Table 15: Nominal and Closed-loop Optimal Costs for $\gamma = 0.8$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	61.05847568
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
23.34280767	19.86228881

Table 16: Nominal and Closed-loop Optimal Costs for $\gamma = 1.0$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.86835686	52.62802941
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
24.36918340	19.36730029

7.1.2 Results obtained with the second mission

This time, the vehicle is constrained to move in a plane, from the given initial position $\boldsymbol{x}_i = [\,0\,\text{ft}\,;\,0\,\text{ft}\,]$ to the fixed final target $\boldsymbol{x}_f = [\,30\,\text{ft}\,;\,10\,\text{ft}\,]$. Also, the vehicle is restricted, initially, to have a heading of $\frac{2\pi}{4}$ rads. The rest of the mission remains unchanged.

7.1.2.1 Open-loop Trajectories

To the difference of the first mission results, this mission allows us to show that there is indeed a difference in the nominal solutions picked by the non-robust and robust policies once the scaling factor γ becomes significantly large. In order to illustrate this, the reader is provided the open-loop solutions to the worst case optimization for varying γ. The resulting trajectories are as presented in Fig. 21 through Fig. 25.

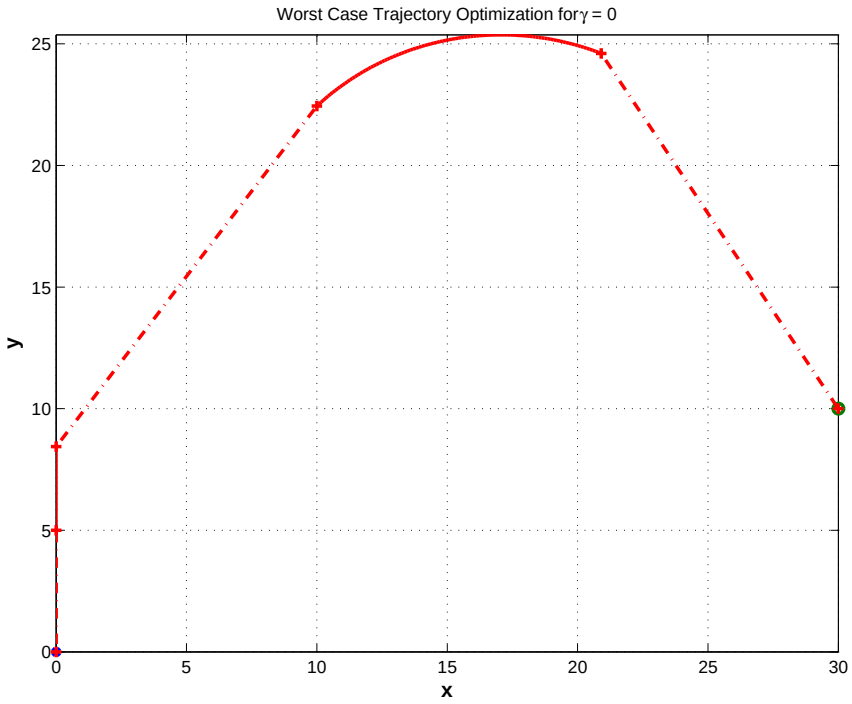


Figure 21: Nominal and Open-loop Solutions to the Robust Approach for $\gamma = 0.0$ - Scenario 2

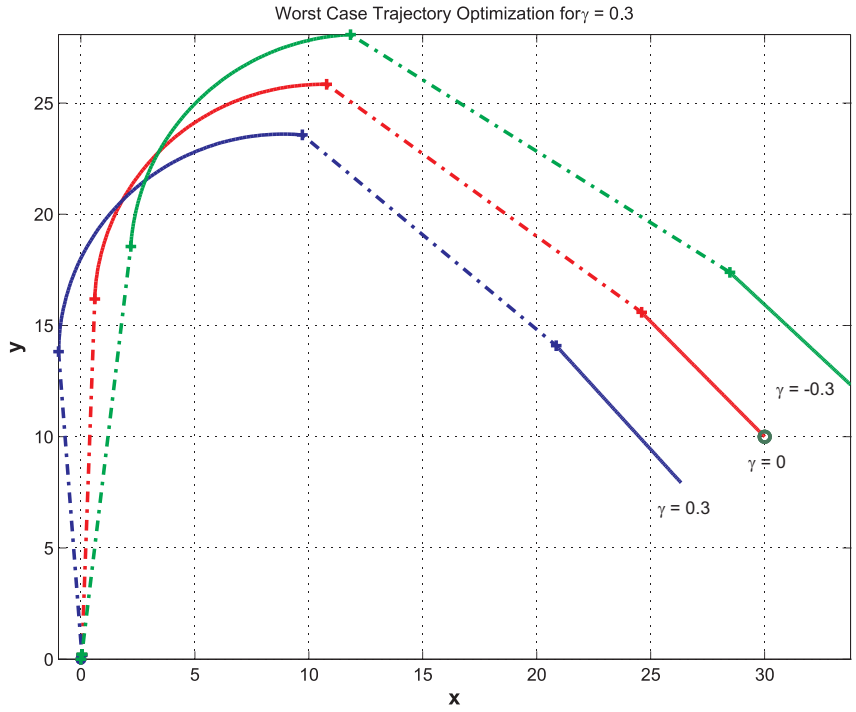


Figure 22: Nominal and Open-loop Solutions to the Robust Approach for $\gamma = 0.3$ - Scenario 2

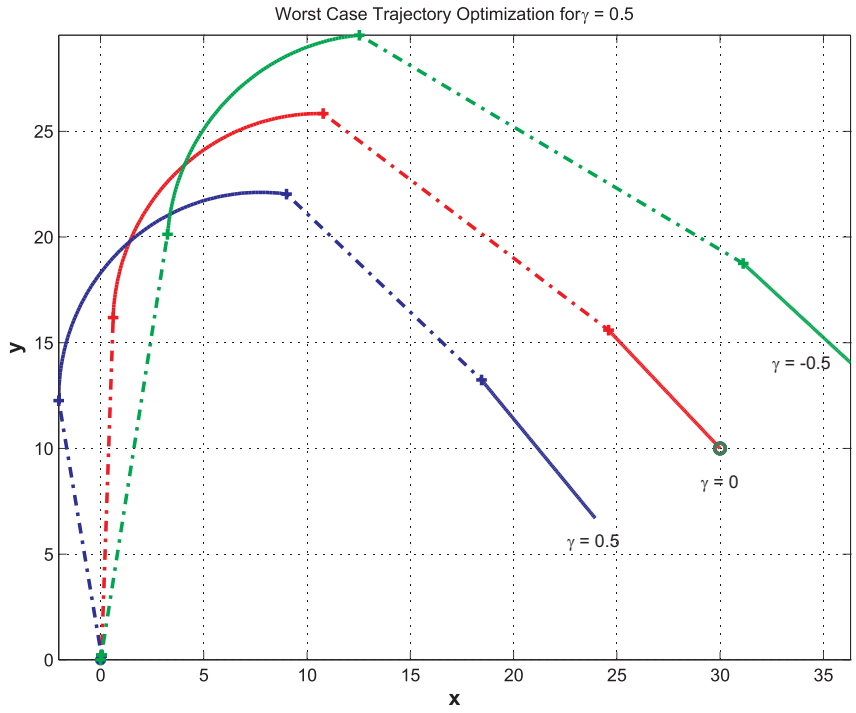


Figure 23: Nominal and Open-loop Solutions to the Robust Approach for $\gamma = 0.5$ - Scenario 2

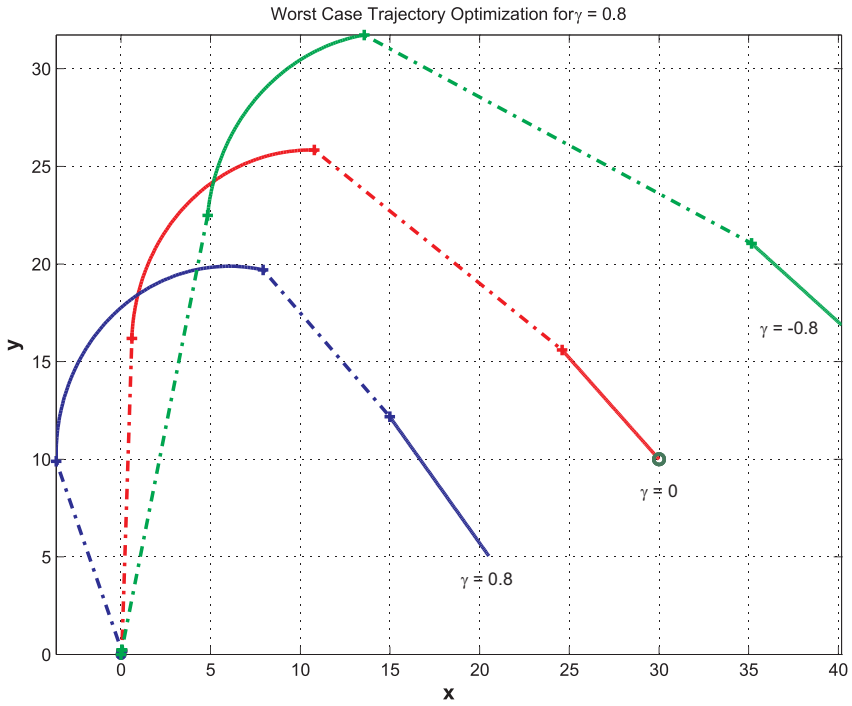


Figure 24: Nominal and Open-loop Solutions to the Robust Approach for $\gamma = 0.8$

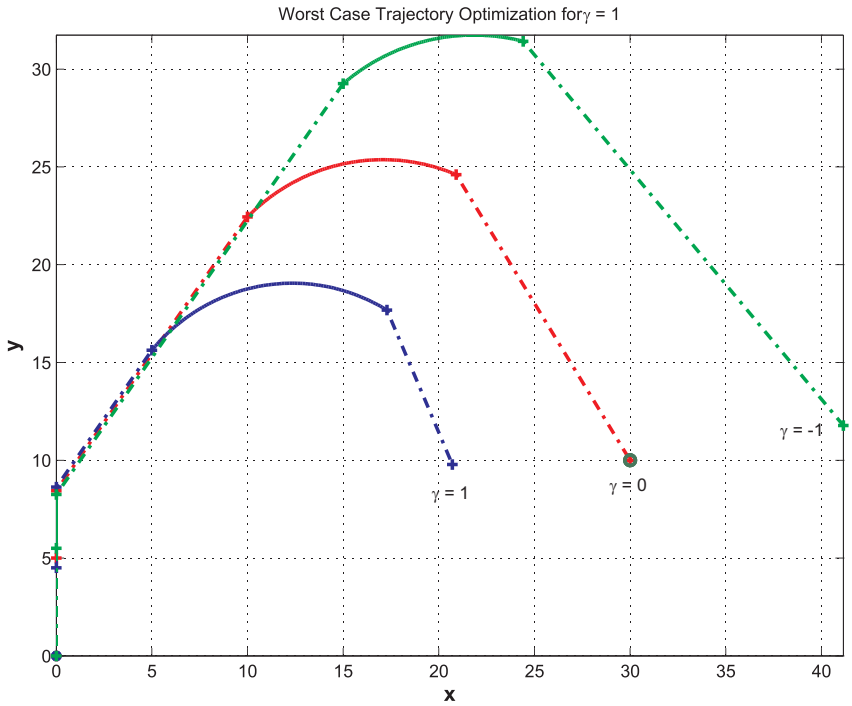


Figure 25: Nominal and Open-loop Solutions to the Robust Approach for $\gamma = 1.0$

Fig. 21 through Fig. 25 suggests that, the worst case scenario approach picks solutions to the optimization problem as a function of the amount of uncertainty that is accounted for, scaled by γ. Moreover, the red curves represent the nominal trajectories associated to $\gamma = 0$. The green and blue trajectories correspond to the open-loop perturbed trajectories for the values of γ given in caption. Moreover, the green curve corresponds to the open-loop curve that results from subtracting all the uncertainties in the library scaled by γ, the blue curve corresponds to the results obtained when all uncertainties in the library are added to the motion primitive parameters, scaled by γ.

7.1.2.2 Optimization Results

As a result to this different mission, similar results are obtained. The tracking performance of the worst case scenario approach is again better than the one for the non-robust policy as predicted. Furthermore, for $\gamma = 1$, i.e., all the uncertainties are accounted for in their entirety, the worst case scenario closed-loop cost is 17.75431320 seconds while the corresponding open-loop optimal cost was 20.31279186 seconds.

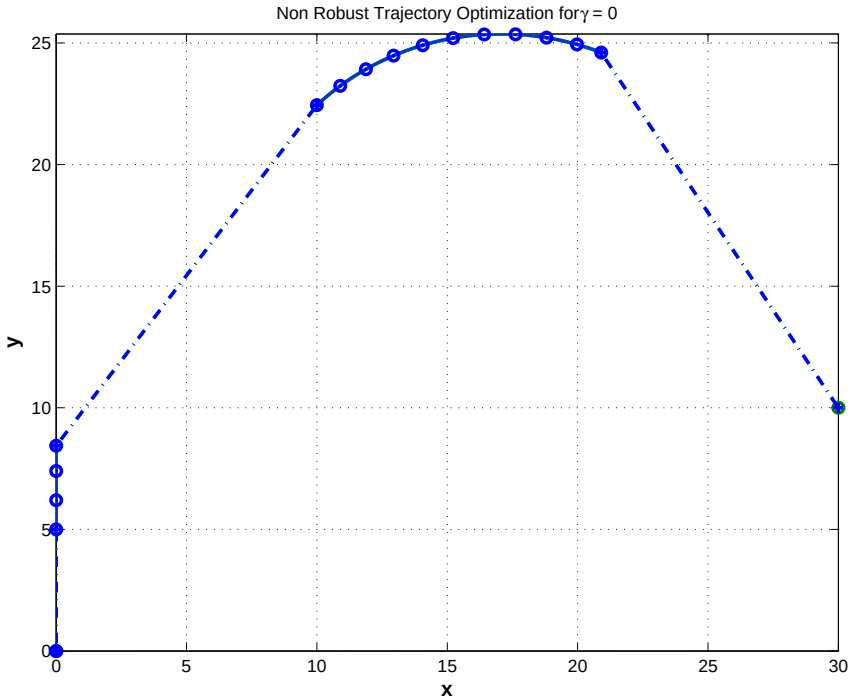


Figure 26: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.0$ - Scenario 2

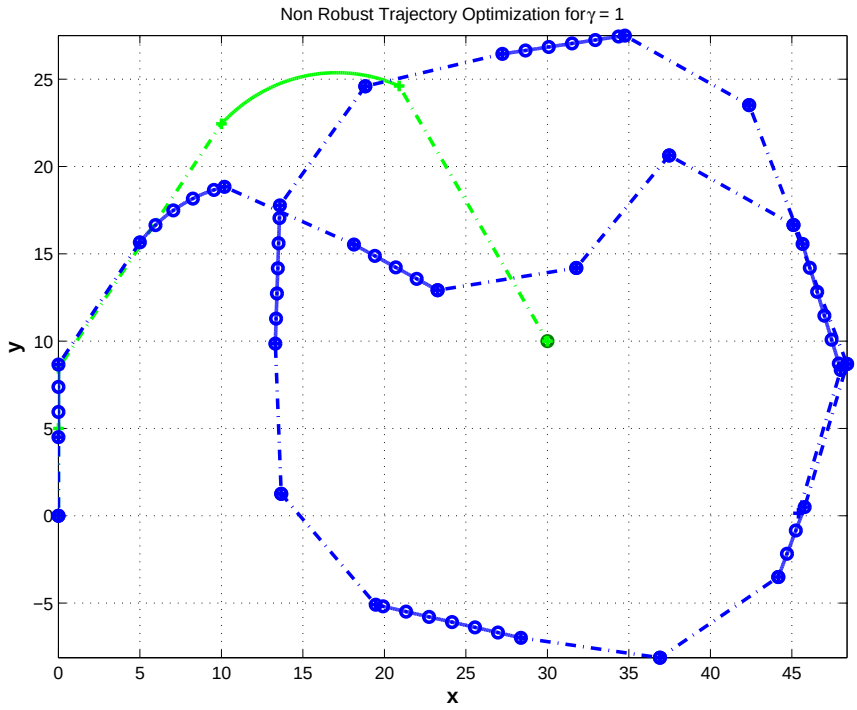


Figure 27: Closed-loop Solution to the Non Robust Approach for $\gamma = 1.0$ - Scenario 2

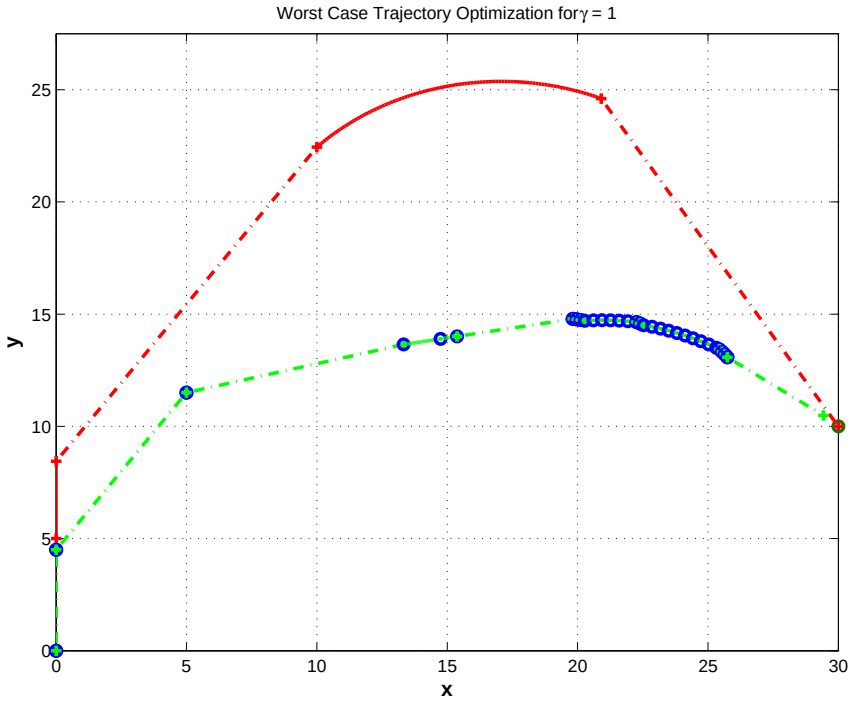


Figure 28: Closed-loop Solution to the Robust Approach for $\gamma = 1.0$ - Scenario 2

Table 17: Nominal and Closed-loop Optimal Costs for $\gamma = 0.0$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.80815399	13.80815399

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.80815399	12.06740119

Table 18: Nominal and Closed-loop Optimal Costs for $\gamma = 1.0$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.80815399	68.52195604
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
20.31279186	17.75431320

7.1.3 Results obtained with the third mission

We finally applied the same methodology to a last set of initial conditions where $\boldsymbol{x}_i = [\,0\,\text{ft}\,;\,0\,\text{ft}\,], \psi_i = \frac{2\pi}{4}$ and $\boldsymbol{x}_f = [\,30\,\text{ft}\,;\,20\,\text{ft}\,]$.

7.1.3.1 Open-loop Trajectories

The open-loop trajectories to this third mission allows the reader to see that the worst case scenario picks solutions that are less prone to the effect of perturbations. The effect, on the solutions picked by the worst case scenario formulation, of uncertainties, was investigated by increasing the scaling factor γ. Fig. 29 and Fig. 30 suggest that, once the scaling factor goes above $\gamma = 0.5$, the worst case scenario formulation will pick solutions that contain motion primitives from the library which are associated to less perturbations. Therefore, as γ increases and hence the amount of uncertainty accounted for increases, the open-loop perturbed trajectories (green and blue curves) become closer to the nominal trajectory.

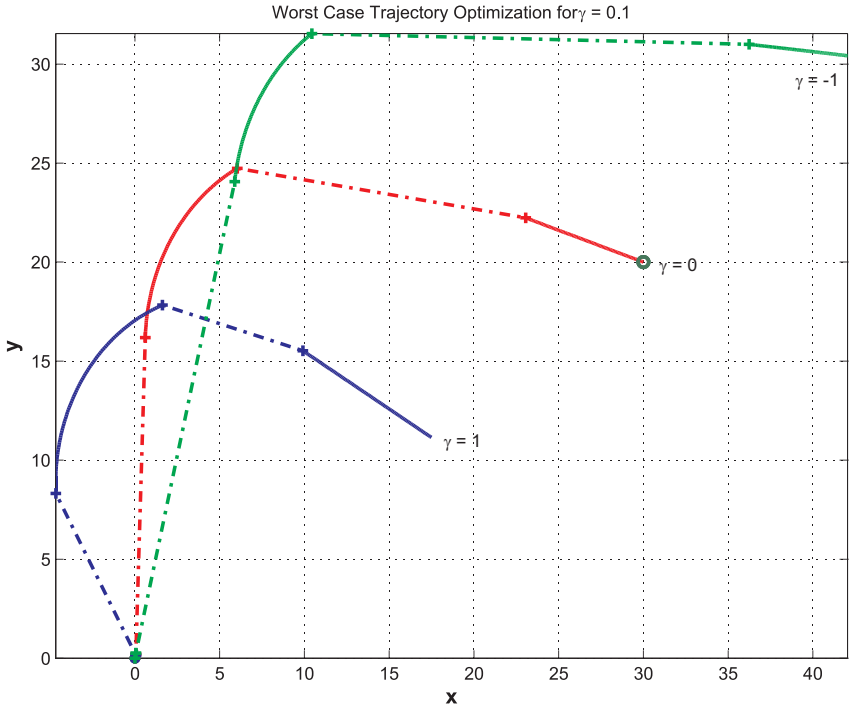


Figure 29: Open-loop solutions to the Robust Approach for $\gamma = 0.1$ - Scenario 3

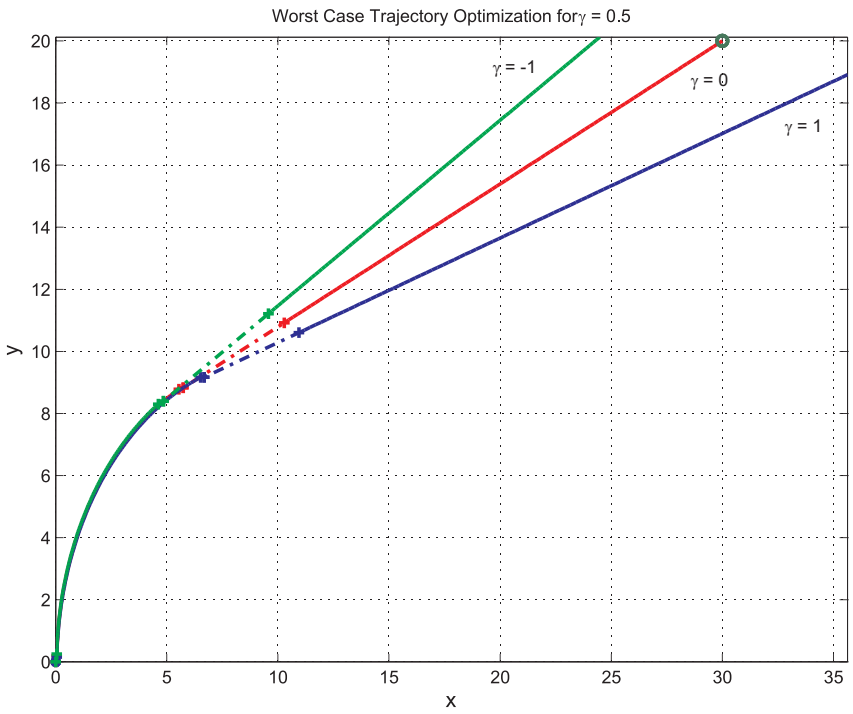


Figure 30: Open-loop solutions to the Robust Approach for $\gamma = 0.5$ - Scenario 3

7.2 Results obtained with the experimental library

Similarly to the first reference library, our methodology was tested on the experimental library. As such, we will be able to show the robustness, to perturbation, of our approach using the dynamic characteristics in terms of primitives of a real vehicle model.

Recall that the experimental library was obtained through simulations performed on a real time flight test verified autonomous vehicle developed here at Georgia Institute of Technology. The motion primitive library relative to the results obtained through simulations on the GT-Max was presented in Chapter 3 in Tables 9 and 10

7.2.1 Results obtained with the first mission

We replicated the same mission that was tested in Section 7.1.1, this time on the experimental library. Motion planning begins at the initial point $\boldsymbol{x}_i = [\,0\,\text{ft}\,;\,0\,\text{ft}\,], \psi_i = \frac{3\pi}{4}$ and the target is defined to be $\boldsymbol{x}_f = [\,30\,\text{ft}\,;\,0\,\text{ft}\,]$.

7.2.1.1 Optimization Results

The behavior of the vehicle, in closed-loop, is indicated in Fig. 31 through Fig. 33 for the non robust formulation of the motion planning problem and Fig. 34 through Fig. 36 for the worst case scenario approach for increasing γ.

Numerical results are presented in Tables 19 through 21 each corresponding to a different scaling factor scenario. In all the scenarios presented here, the closed-loop robust solutions are able to reach the target in lesser time than the corresponding non-robust solutions. As means of comparison, the nominal open-loop solution to the non-robust optimization reaches the target after 17.196 seconds while it fails to hit the target, or come to the allowed vicinity of it in closed-loop for a scaling factor of 0.7. However, the worst case scenario solution arrives to the target after 24.311 seconds for the same scaling factor.

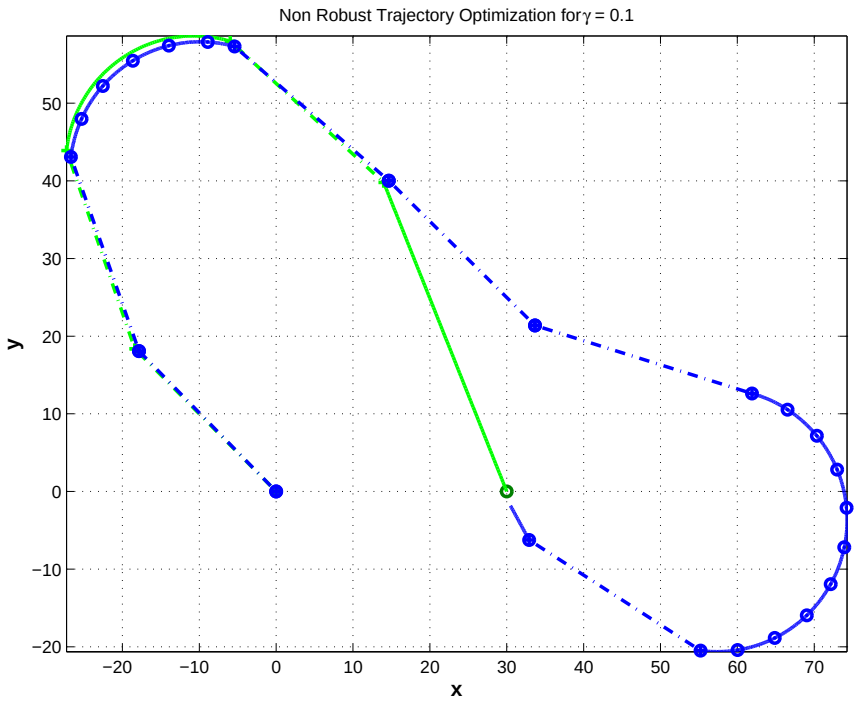


Figure 31: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.1$ - Scenario 1

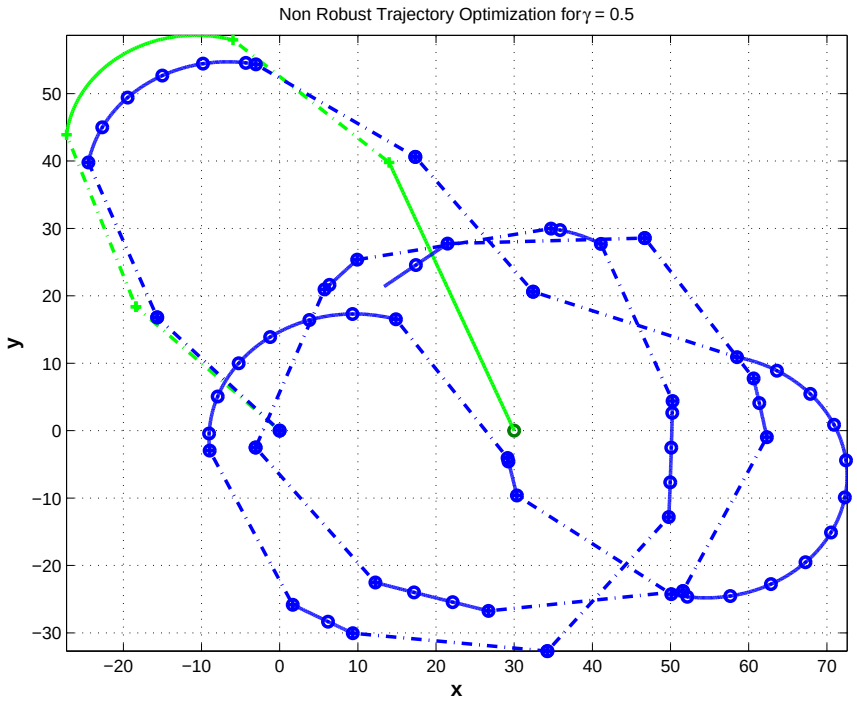


Figure 32: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.5$ - Scenario 1

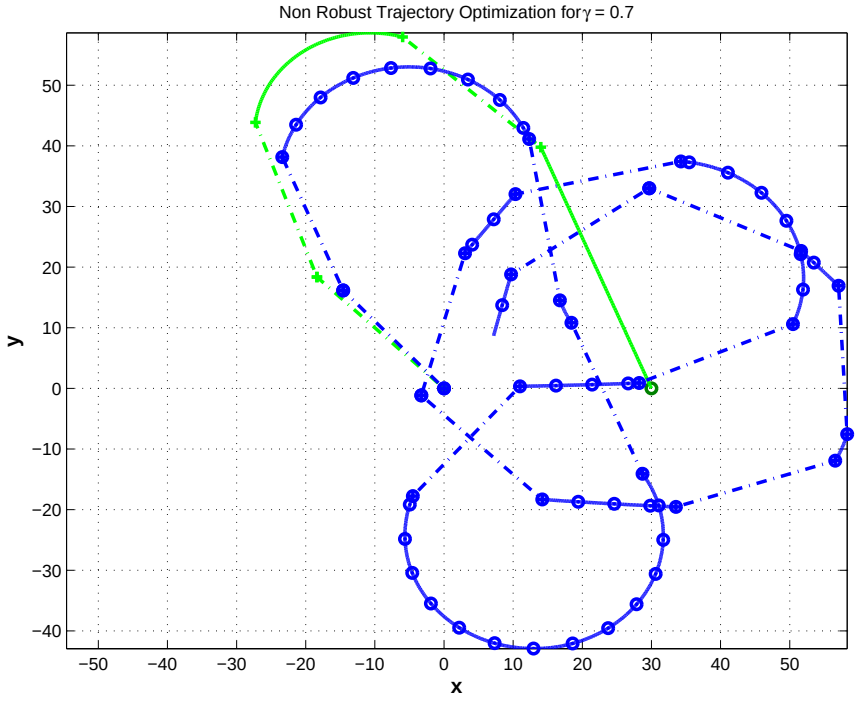


Figure 33: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.7$ - Scenario 1

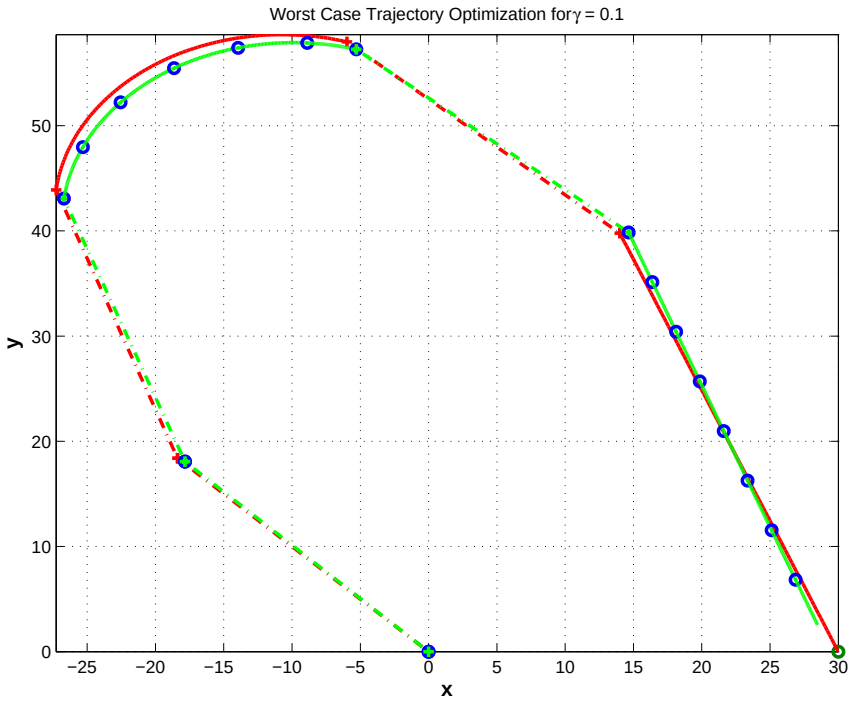


Figure 34: Closed-loop Solution to the Robust Approach for $\gamma = 0.1$ - Scenario 1

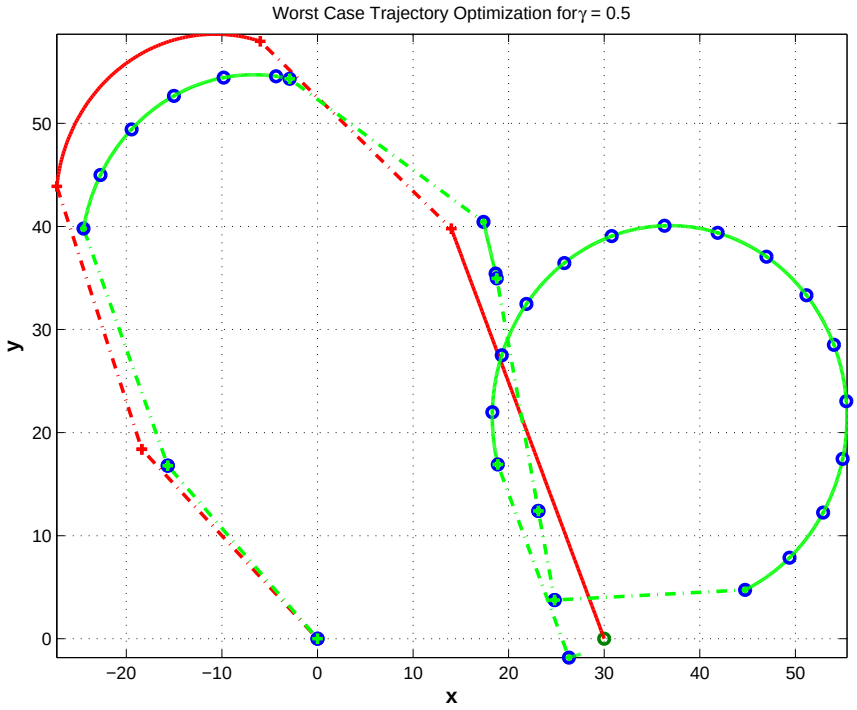


Figure 35: Closed-loop Solution to the Robust Approach for $\gamma = 0.5$ - Scenario 1

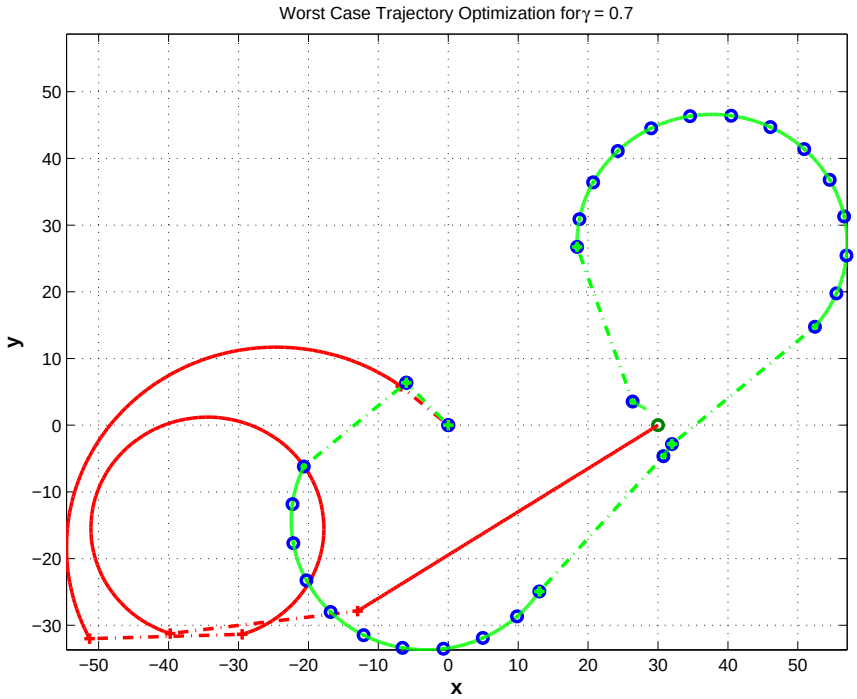


Figure 36: Closed-loop Solution to the Robust Approach for $\gamma = 0.7$ - Scenario 1

Table 19: Nominal and Closed-loop Optimal Costs for $\gamma = 0.1$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
17.19564465	26.75174872

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
17.58223420	16.05815770

Table 20: Nominal and Closed-loop Optimal Costs for $\gamma = 0.5$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
17.19564465	65.37049147
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
19.12782333	32.03125826

Table 21: Nominal and Closed-loop Optimal Costs for $\gamma = 0.7$ - Scenario 1

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
17.19564465	52.68776743
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
60.51056468	24.31112894

It is of interest at this point to compare the nature of maneuvers and trim conditions that were picked, as solutions to both non-robust and worst case scenario policies. To this end, Figures 37 – 48 provide the distribution, as pie charts of occurring maneuvers and trim conditions as well as uncertainties present in those chosen characteristics.

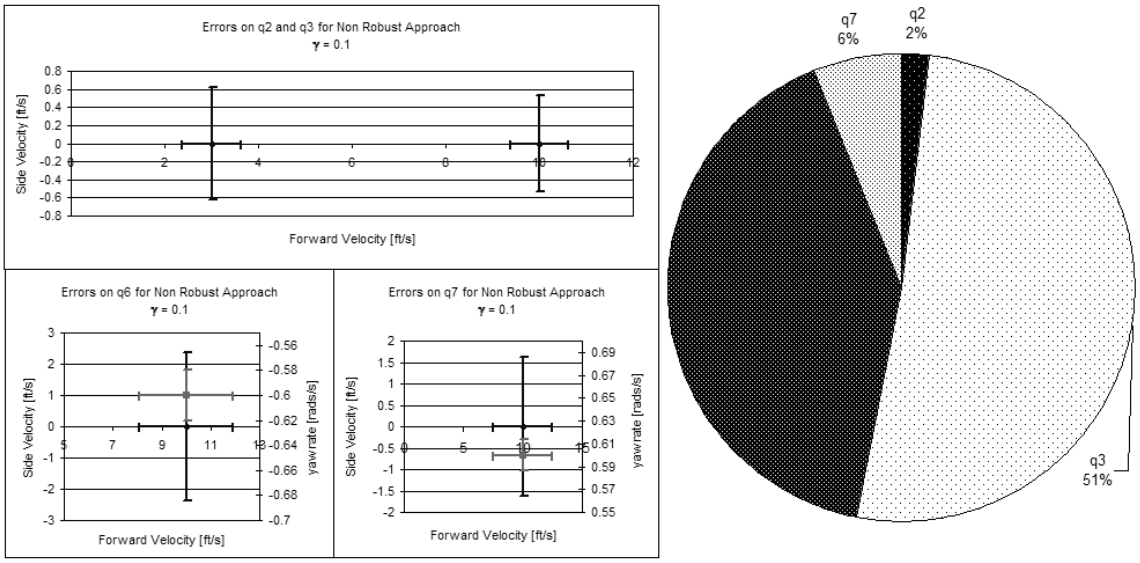


Figure 37: Trim Occurrences for Non Robust Approach $\gamma = 0.1$

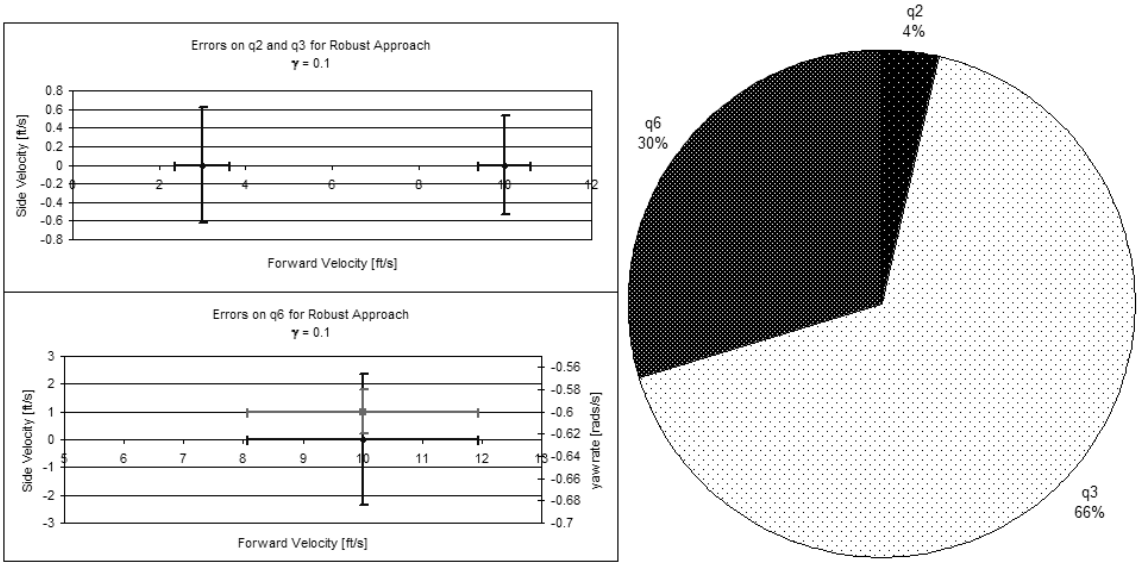


Figure 38: Trim Occurrences for Robust Approach $\gamma = 0.1$

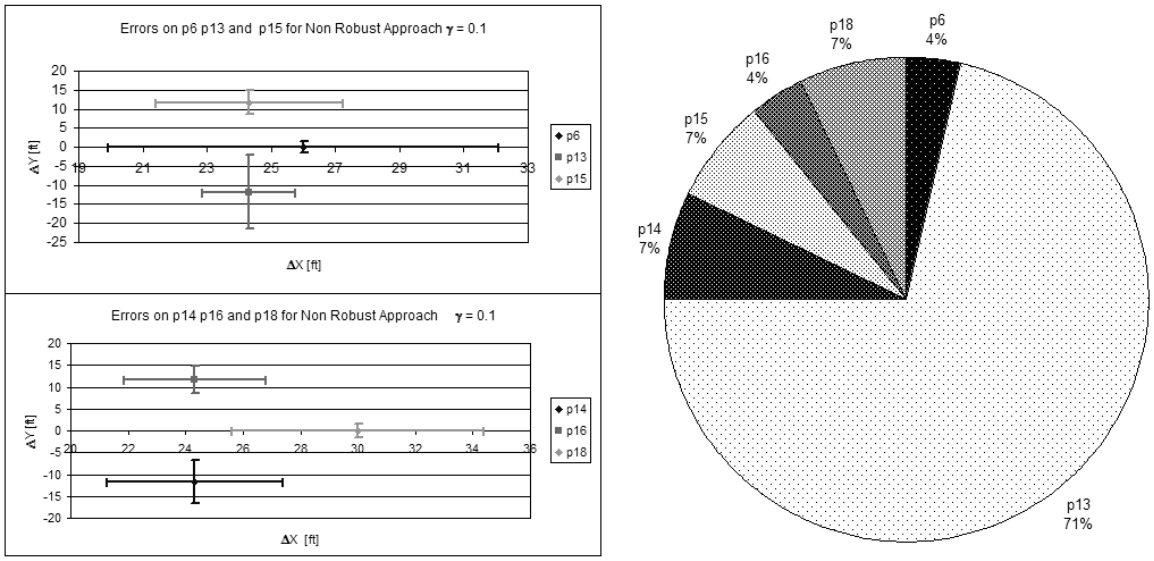


Figure 39: Maneuver Occurrences for Non Robust Approach $\gamma = 0.1$

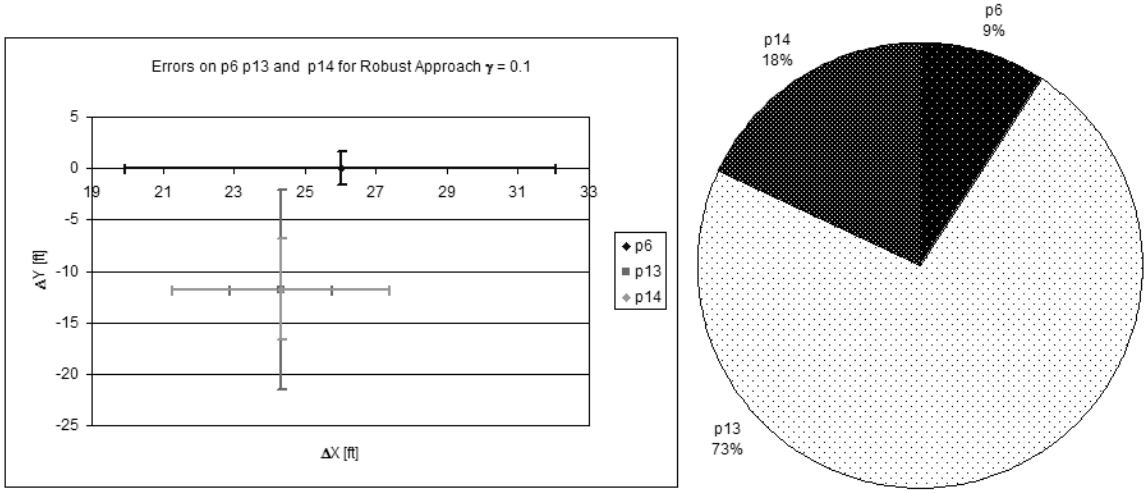


Figure 40: Maneuver Occurrences for Robust Approach $\gamma = 0.1$

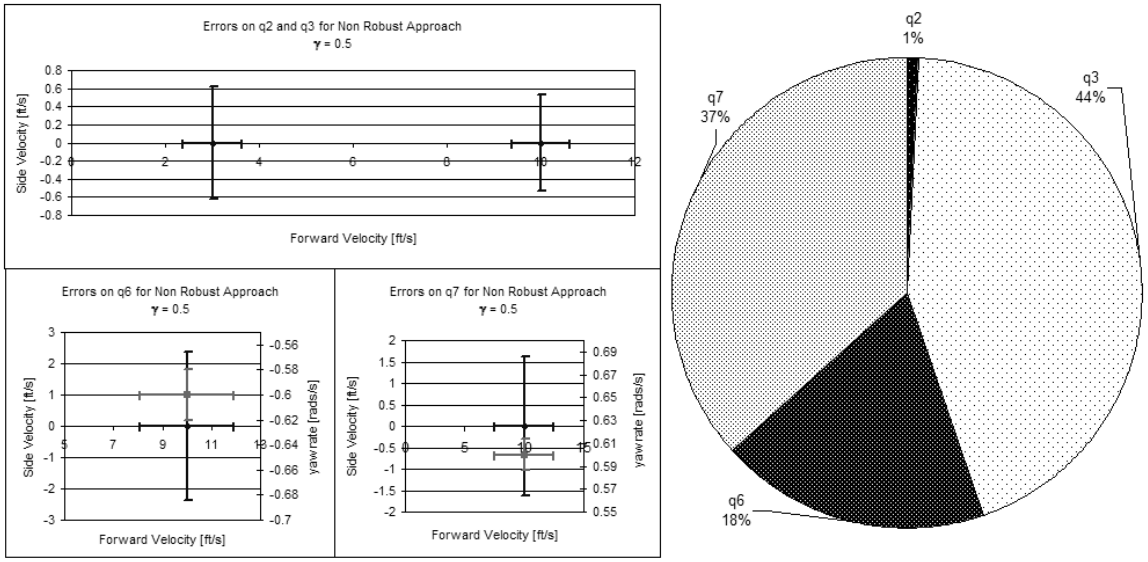


Figure 41: Trim Occurrences for Non Robust Approach $\gamma = 0.5$

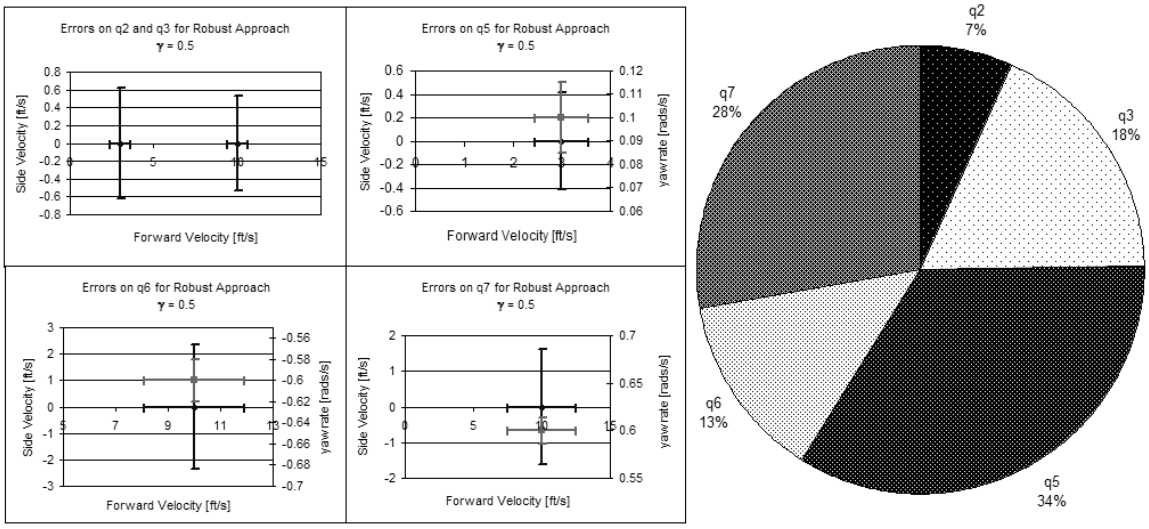


Figure 42: Trim Occurrences for Robust Approach $\gamma = 0.5$

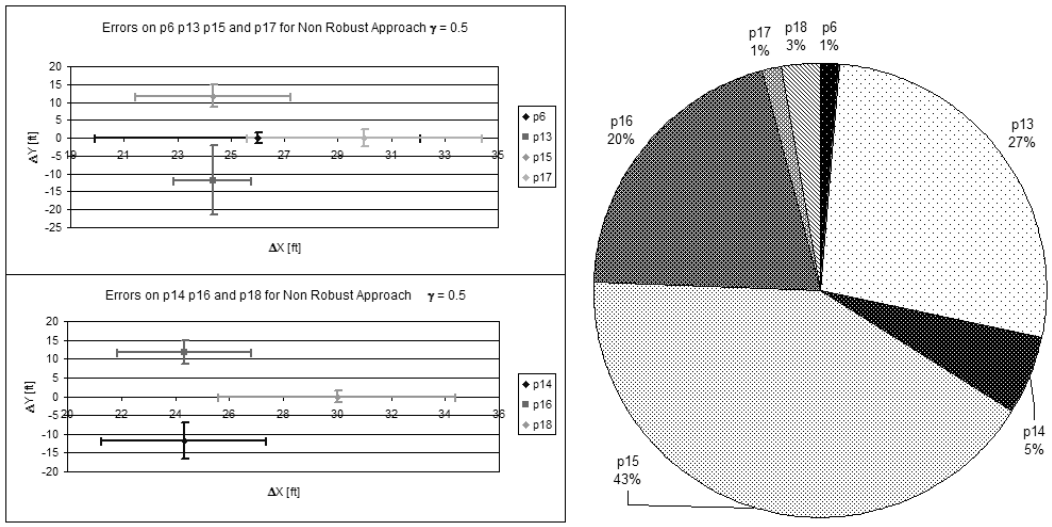


Figure 43: Maneuver Occurrences for Non Robust Approach $\gamma = 0.5$

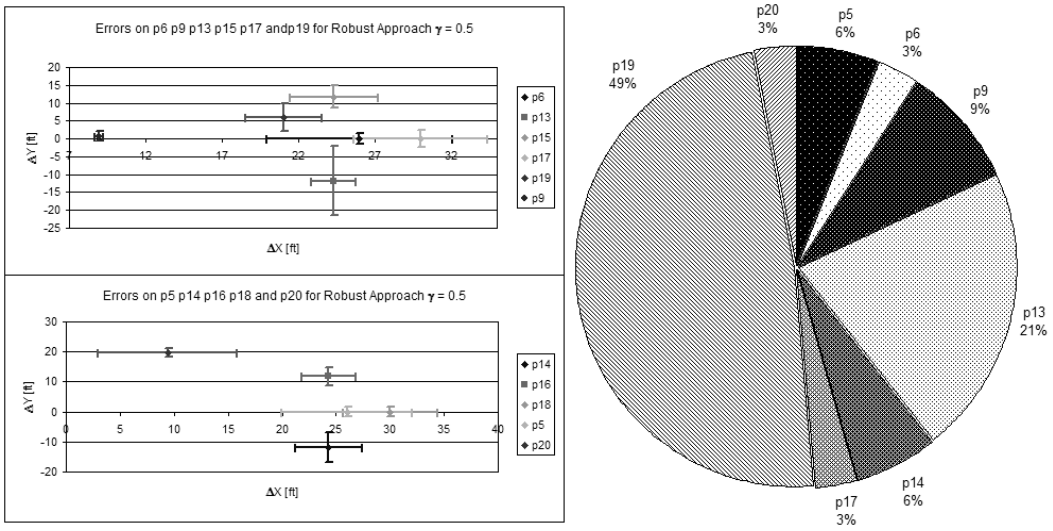


Figure 44: Maneuver Occurrences for Robust Approach $\gamma = 0.5$

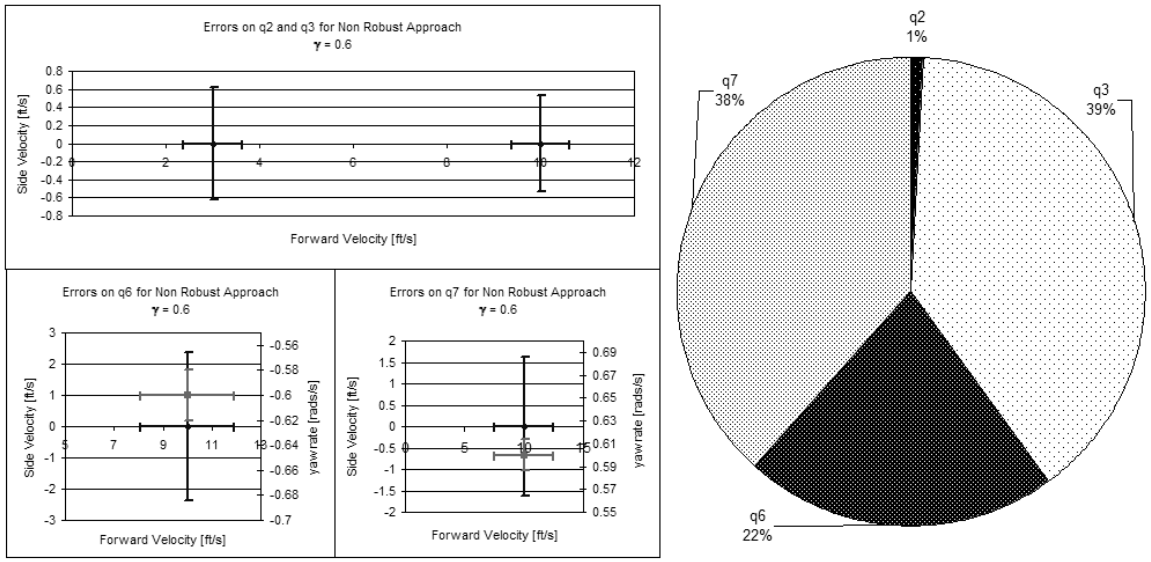


Figure 45: Trim Occurrences for Non Robust Approach $\gamma = 0.6$

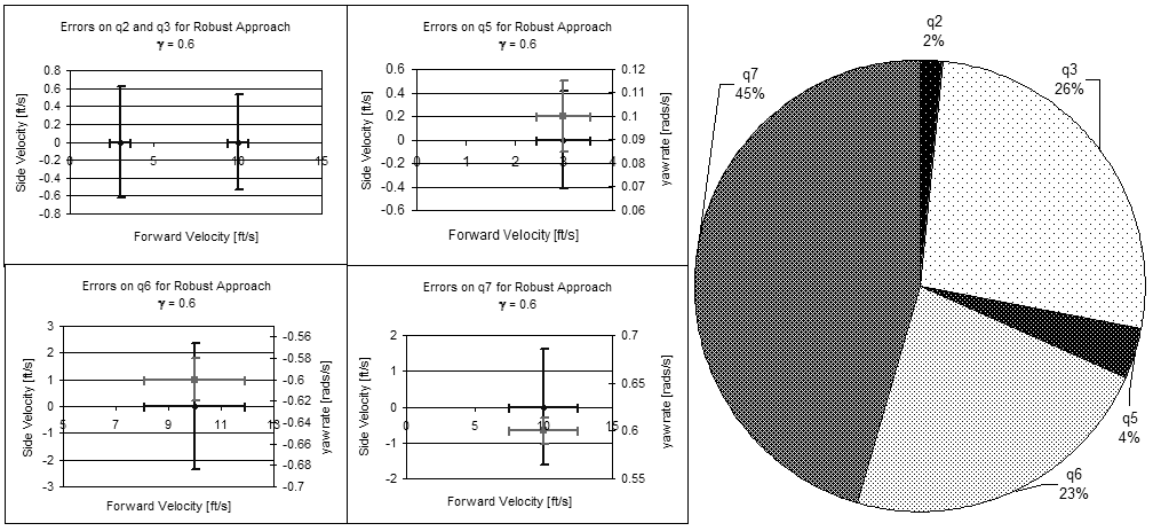


Figure 46: Trim Occurrences for Robust Approach $\gamma = 0.6$

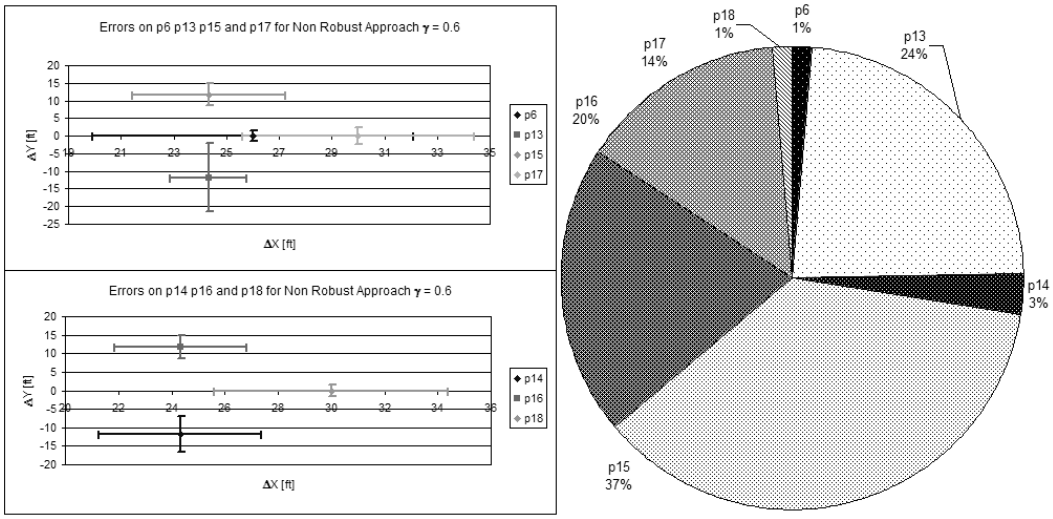


Figure 47: Maneuver Occurrences for Non Robust Approach $\gamma = 0.6$

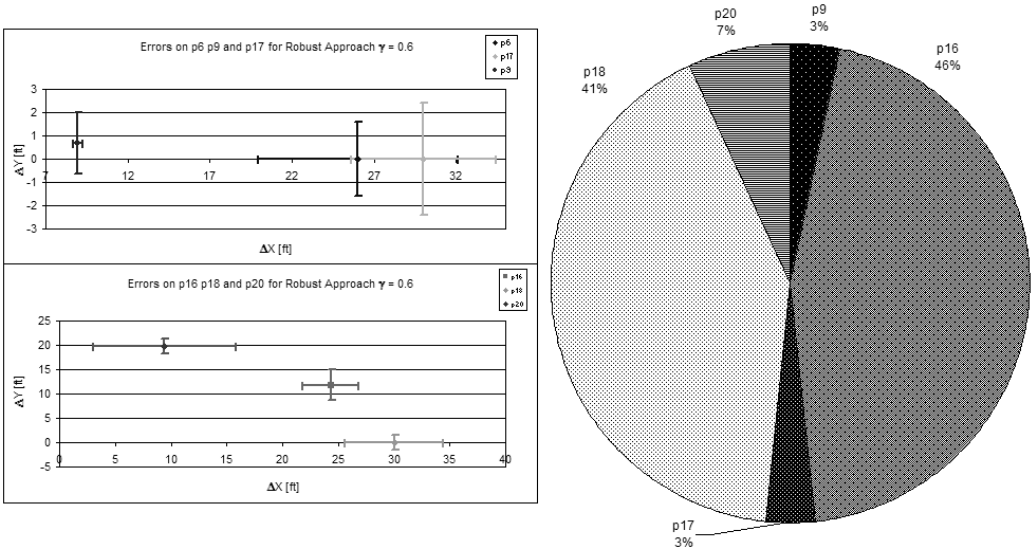


Figure 48: Maneuver Occurrences for Robust Approach $\gamma = 0.6$

The above presented figures suggest that, as γ increases, the robust policy start picking solutions that contain trim number 5, which is a slow right turn. For example in Fig. 42, the robust solutions contain, 36% of the time, trim number 5. However, the non robust policy uses both straight and turning flight trim conditions that are associated with large velocities.

As far as maneuvers are concerned, referring to Fig. 55, the robust policy chooses optimal sequences of trims and maneuvers that contain maneuver 19, which is the maneuver that permits the vehicle to slow down, from a fast right turn to a fast left turn.

7.2.2 Results obtained with the second mission

This time the vehicle is constrained to move from the initial point $\boldsymbol{x}_i = [\,0\,\text{ft}\,;\,0\,\text{ft}\,], \psi_i = \frac{2\pi}{4}$ to the target $\boldsymbol{x}_f = [\,50\,\text{ft}\,;\,20\,\text{ft}\,]$.

7.2.2.1 Optimization Results

The behavior of the vehicle, in closed-loop, is indicated in Fig. 49 through Fig. 53 and Fig. 54 through Fig. 58 for increasing γ.

Numerical results are presented in Tables 22 through 26 each corresponding to a different scaling factor scenario. As for the previous scenario, the closed-loop robust solutions are able to reach the target in lesser time than the corresponding non-robust solutions. For small values of γ, tracking of the trajectory is also better than the non-robust solutions.

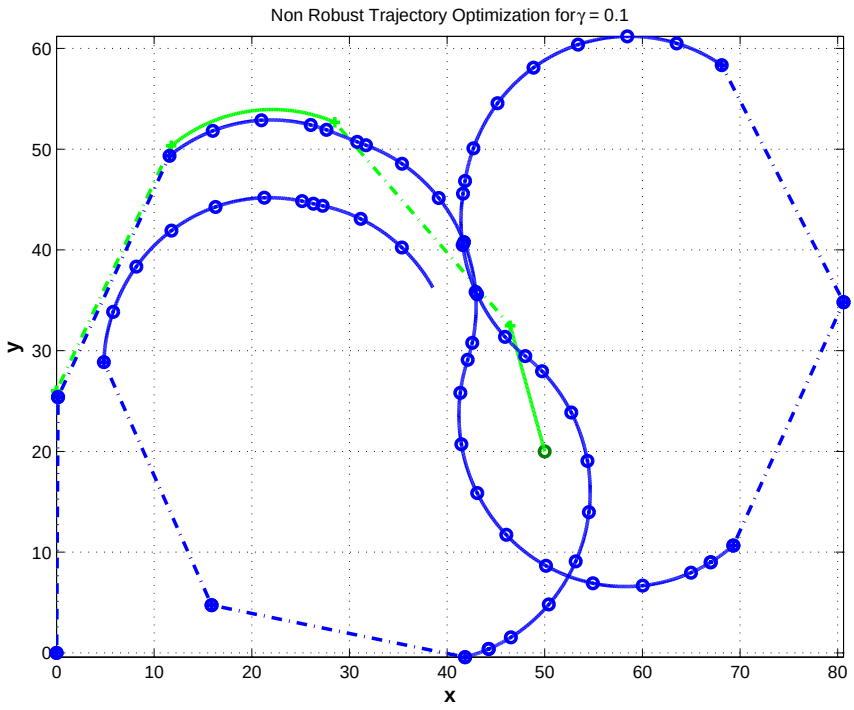


Figure 49: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.1$ - Scenario 2

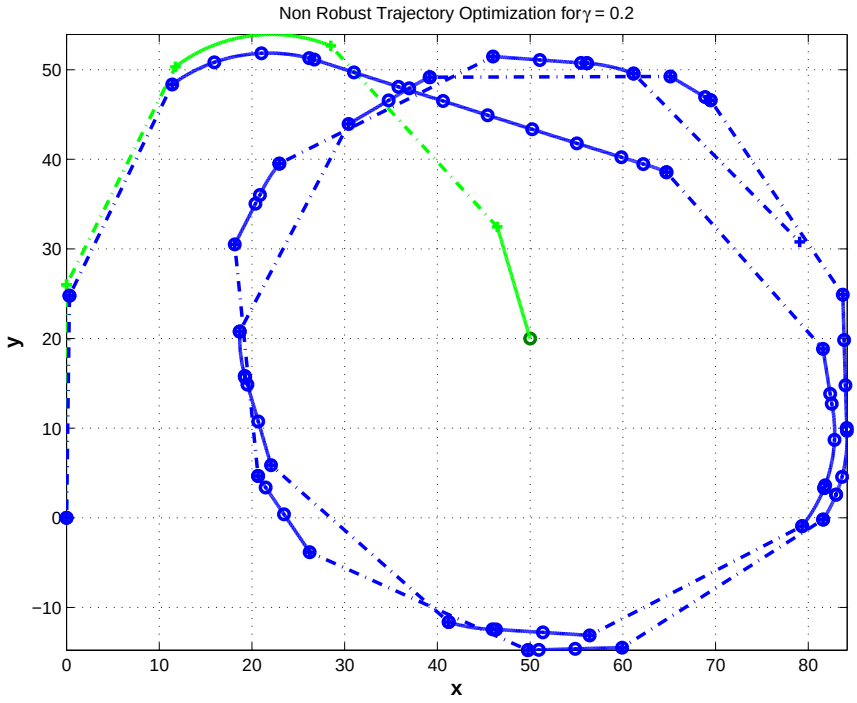


Figure 50: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.2$ - Scenario 2

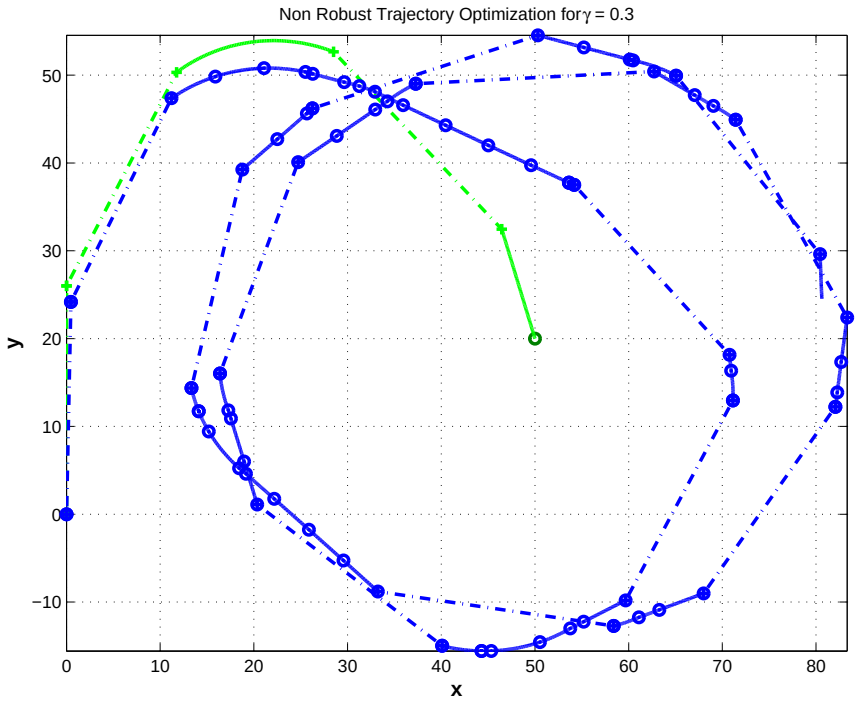


Figure 51: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.3$ - Scenario 2

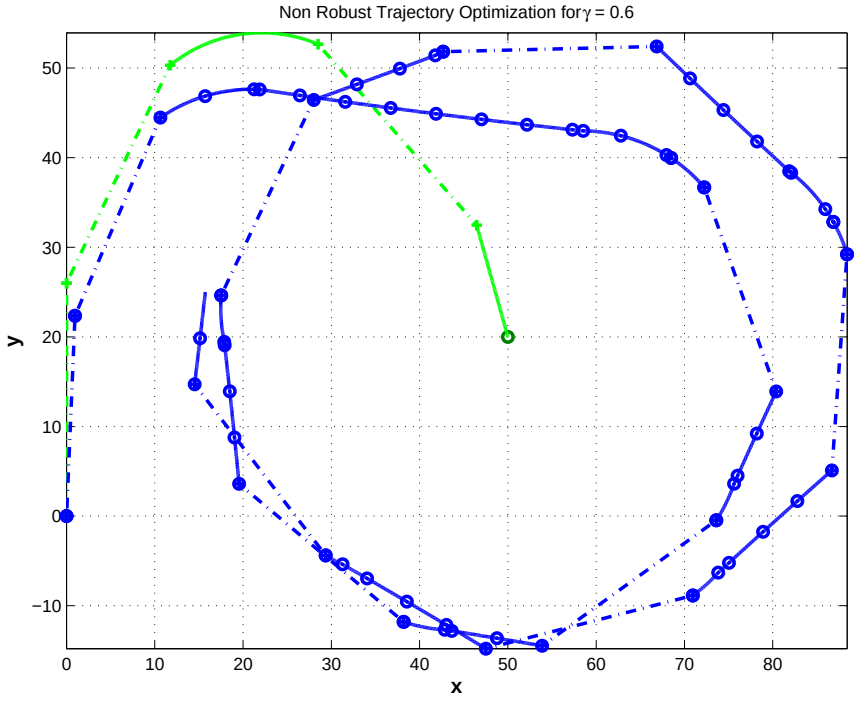


Figure 52: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.6$ - Scenario 2

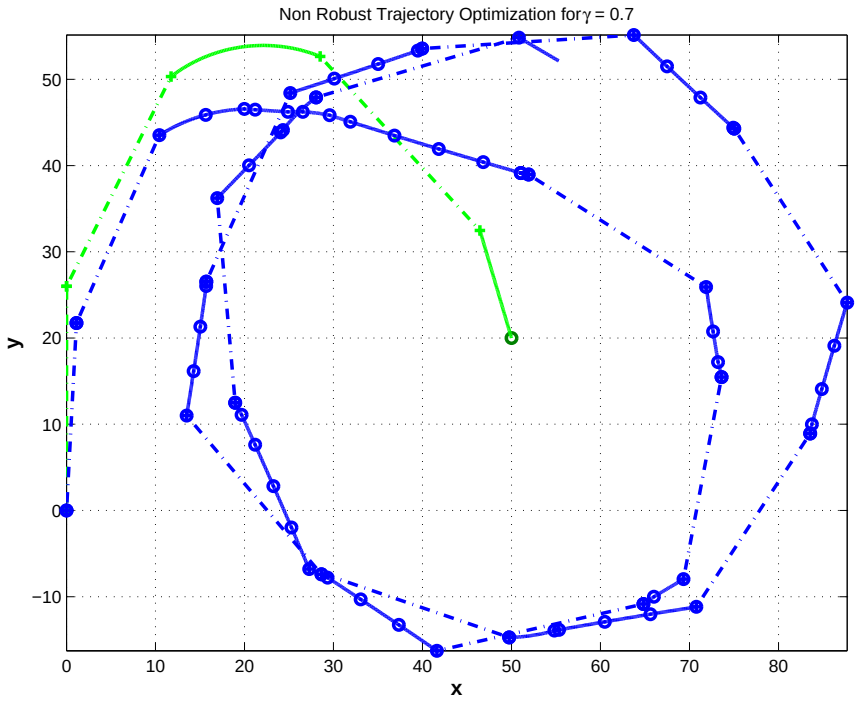


Figure 53: Closed-loop Solution to the Non Robust Approach for $\gamma = 0.7$ - Scenario 2

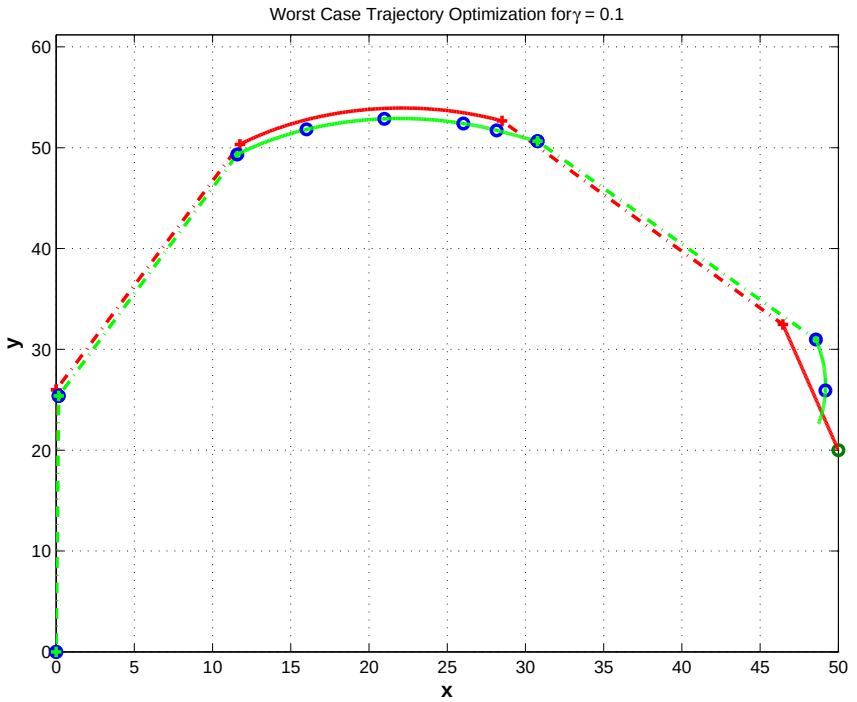


Figure 54: Closed-loop Solution to the Robust Approach for $\gamma = 0.1$ - Scenario 2

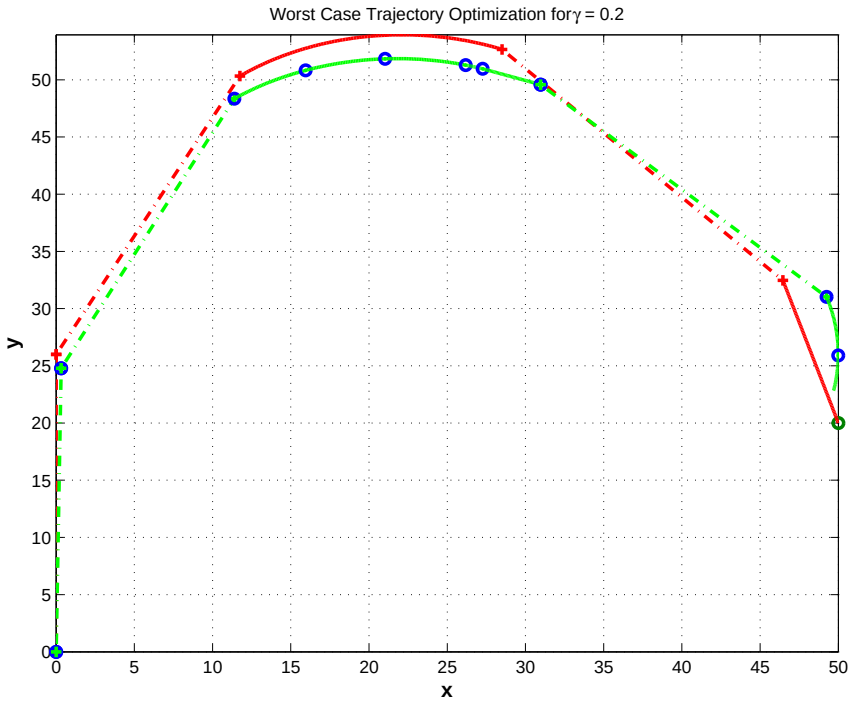


Figure 55: Closed-loop Solution to the Robust Approach for $\gamma = 0.2$ - Scenario 2

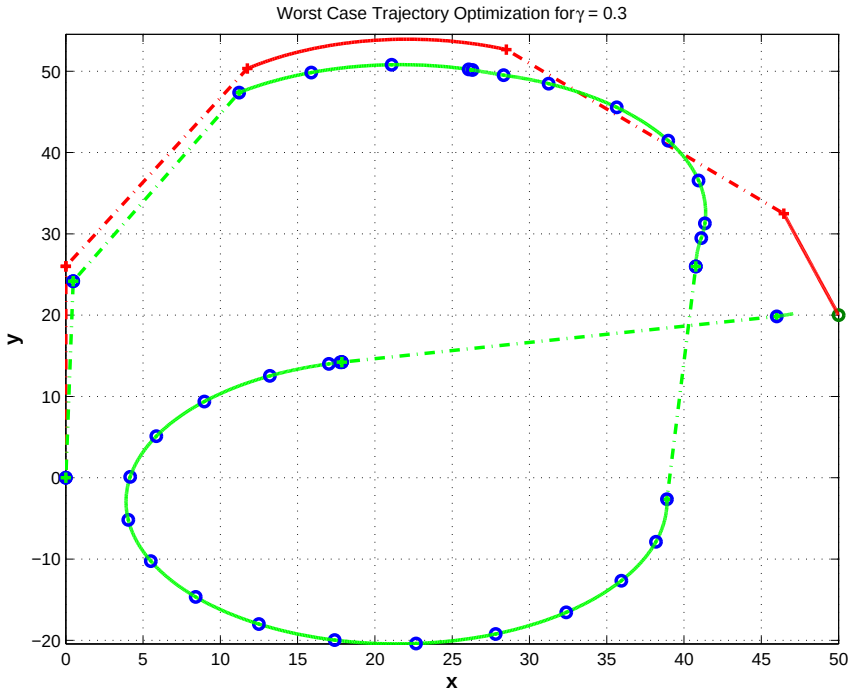


Figure 56: Closed-loop Solution to the Robust Approach for $\gamma = 0.3$ - Scenario 2

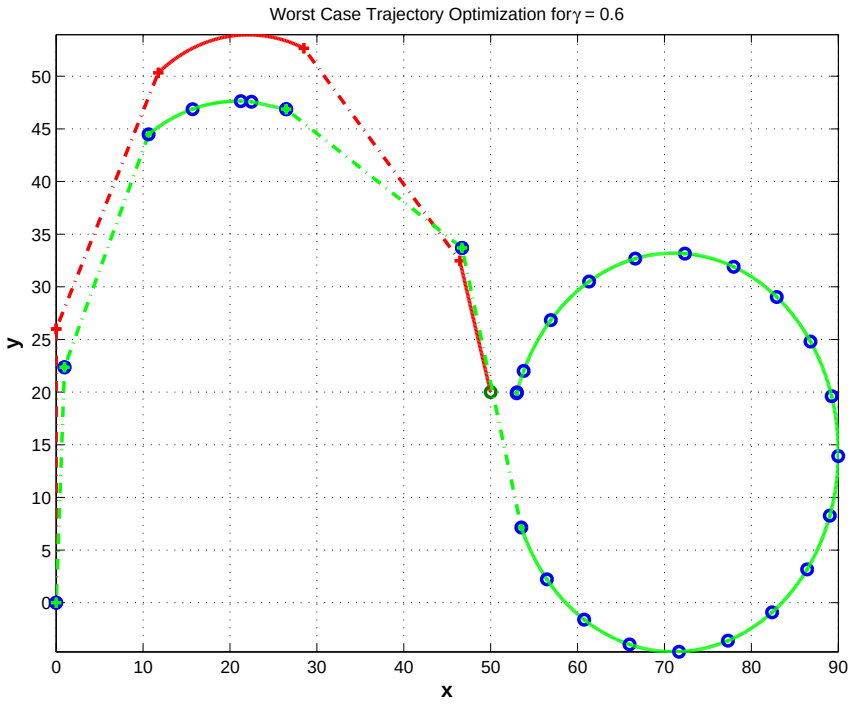


Figure 57: Closed-loop Solution to the Robust Approach for $\gamma = 0.6$ - Scenario 2

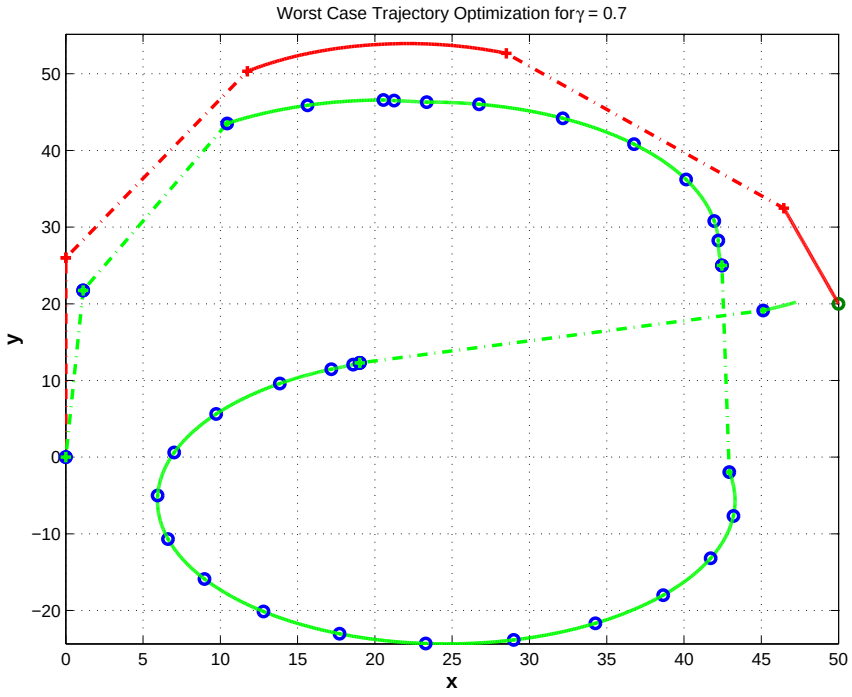


Figure 58: Closed-loop Solution to the Robust Approach for $\gamma = 0.7$ - Scenario 2

Table 22: Nominal and Closed-loop Optimal Costs for $\gamma = 0.1$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.07107247	38.77440384
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.42410657	11.98088517

Table 23: Nominal and Closed-loop Optimal Costs for $\gamma = 0.2$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.07107247	53.02731146
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.77880971	11.79107632

Table 24: Nominal and Closed-loop Optimal Costs for $\gamma = 0.3$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.07107247	51.42835246
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
14.13470080	22.47452309

Table 25: Nominal and Closed-loop Optimal Costs for $\gamma = 0.6$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.07107247	43.98039397
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
15.20568435	21.05150780

Table 26: Nominal and Closed-loop Optimal Costs for $\gamma = 0.7$ - Scenario 2

(a) Non Robust Solutions

Nominal Cost [s]	Real Cost [s]
13.07107247	45.77119840
Maximum Number of Iteration Reached	

(b) Robust Solutions

Nominal Cost [s]	Real Cost [s]
15.56276175	20.41190763

CHAPTER VIII

CONCLUSIONS

8.1 Concluding Remarks

The focus of this research was to develop an algorithm that would enable an autonomous vehicle to compute a motion plan to reach a target in minimum time, given a set of initial conditions and given uncertain parameters affecting its dynamics. The proposed solution of the motion planning problem was designed for a time-invariant dynamic control system with symmetries: the Maneuver Automaton [12]. With this representation, motion plans were defined as a concatenation of trajectory primitives, relative equilibria and maneuver conditions, that were defined within a motion primitive library. The motion planning problem was addressed through the use of optimal control theory and the constraints to the problem were defined on the states and controls. The solution to the time optimal problem addressed through the use of the Maneuver Automaton leads an approximation of real trajectories and therefore, the solutions that were computed according to the described method are, in general, not optimal. However, the usage of this automaton results in a reduction in computational complexity, as the quantization of the states and controls is done off-line and stored within a library. Therefore there is no need for time, state or control quantization.
In order to address the inevitable presence of uncertainties in the environment surrounding the vehicle and in its dynamics, two motion planning problems, one that omits and another that accounts uncertainties, were defined. Uncertainties were accounted for, both in the open-loop computation of an optimal guidance cost function, time for instance, and in a closed-loop, for the formulation we named worst case approach. The resulting worst case solution therefore represents a trade-off between time to reach the goal and uncertainties present in the primitives minimization. Two different Maneuver Automaton representations, one based on kinematic evaluation of motion primitives and the other relying on data

of an existing dynamic system were defined.

The examples provided for both libraries highlighted, through the comparison of both non-robust and worst case scenario policies introduced in this work, that the addition of uncertainties within the optimization problem results in closed-loop trajectories that are more robust to perturbations and whose time performance is better. These results were validated for both the theoretical and the experimental library and therefore reinforce the initial motivation of the research, which is to guarantee the overall robustness of the control system. The nominal motion planning problem, which omits uncertainties in the system parameters may lead to the failure of the mission, whereas the worst case motion planning problem, in which uncertainties are introduced within the tactical layer, may result in the accomplishment of mission objectives.

Moreover, uncertainties are statistical quantities and are therefore unknown at a given instant in time. As a result, experiments at different level of uncertainties were run to back up the results we obtained.

8.2 Problems Encountered

8.2.1 Suboptimality Issues

One of the drawbacks of the maneuver automaton framework is related to the limited number of actions one is constrained to perform given a finite set of trims and maneuvers defining the library. It follows that motion plans constructed through the concatenation of suitable primitives are approximations of real trajectories and hence, the are sub-optimal.

8.2.2 Initial Guess Computations

As for any nonlinear programming problem, the computation of initial guesses was highly complex. Even thought the optimizer was given a smart guess for initial values of coasting times, it was sometimes confronted to infeasibility problems due to poor initial guesses.

8.2.3 Decent Motion Primitive Library

At the core of the Maneuver Automaton framework lies the motion primitive library. This library depends on the dynamic characteristics and controller architecture of the system it

issues from. Obtaining this representation for a given system therefore requires statistical evaluations of not only relative equilibria and transitions as well as associated errors in those parameters. Extending the library to a greater number of primitives is therefore very time consuming.

8.2.4 Modeling Uncertainties

Dr. Eric N. Johnson and his team have been doing such a good job with the controller of the GT-Max that it already deals very well with uncertainties and hence leaves very small errors and uncertainties to work with. This was a problem in the generation of a good library for the purposes of this research since the approach relies greatly on the nature and quantity of uncertainties present in trims and maneuvers.

8.3 Future Work

In the presented work, we have introduced a new methodology, which enables the incorporation of uncertainties into the motion planning of UAVs. We have also investigated the effect of increasing the factor that scales the uncertainties present in the system primitives. The results provided in this work suggest that as γ becomes larger and larger, the deviation for both the non-robust and worst case approaches becomes greater. As such the system tends to behave inadequately.

It would be of interest to optimize the amount of uncertainty that one can incorporate within the library of motion primitives, which will still ensure a satisfactory behavior of the UAV. The motivation under the development of this third optimization is the brief knowledge one has in reality, on the uncertainties present in the motion primitives. In fact, a detailed knowledge on the uncertainties, as incorporated into the library so far, can be a limiting factor. For example, the effect of environmental uncertainties, such as wind gust or turbulence, on a Boeing 747 and on a UAV is considerably different. Therefore, the uncertainties in the system parameters due to the presence of gust or turbulence, will differ from system to system.

In more general cases, it will be desired to trade some performance, as introduced in Chapter 4, as measured by the performance index $J(\boldsymbol{x}, \boldsymbol{u})$, to gain in robustness. More precisely,

we will be satisfied with any value of the performance index J such that

$$J_{opt} \leq J \leq \alpha J_{opt} \quad \alpha \geq 1$$

where α measures the acceptable performance loss.

Under this perspective, we will consider the family of uncertain parameters within the motion primitives as bounded by an unknown value $\gamma > 0$. Increasing γ indicates that the uncertainty accounted for is greater with respect to the nominal value of the system parameters.

Robust optimal control maximizes the uncertainty level γ that still guarantees a certain acceptable performance decrease with respect to the non-robust case, i.e. it finds the maximum uncertainty level that satisfies the trade-off we are ready to accept.

The robust cost function is identical to the worst case scenario cost function,

$$J^R(\boldsymbol{q}, \boldsymbol{\tau}, \gamma) = \sum_{i=1}^{n_T} \tau_i + \sum_{i=1}^{n_M} (T_i + \gamma \epsilon_{T_i})$$

however , the uncertainty level is added to the optimization, which leads to the redefinition of the optimal control problem for the robust case.

Problem 8.1. *Given the maneuver automaton in consideration, an input position pair* $\boldsymbol{x}_i, \boldsymbol{x}_f \in X$*, find a hybrid control input pair* $\{\boldsymbol{q}, \boldsymbol{\tau}\}$ *such that,*

$$\boldsymbol{x}_{2k} = \phi((\boldsymbol{w}_{q_k} + \gamma \epsilon_{\boldsymbol{w}_{q_k}})\tau_k)\, \boldsymbol{x}_{2k-1} \qquad \textit{in a trim condition} \qquad k = 1, ..., n_T \tag{9a}$$

$$\boldsymbol{x}_{2k+1} = \boldsymbol{x}_{2k} + \Delta \boldsymbol{x}_p + \gamma \epsilon_{\Delta \boldsymbol{x}_p} \qquad \textit{in a maneuver} \qquad k = 1, ..., n_M \tag{9b}$$

$$\|\boldsymbol{x}_{2n_T} - \boldsymbol{x}_f\| \leq \delta \;\; \textit{where}\, \delta \,\textit{is a tolerance} \qquad \textit{on the final position error} \tag{9c}$$

$$\min_{\boldsymbol{q} \in \mathscr{U}_T} \; \min_{\boldsymbol{\tau}} \; \max_{\gamma} J^R(\boldsymbol{q}, \boldsymbol{\tau}, \gamma) \qquad \textit{where}\, \gamma \in [-1; 1] \tag{9d}$$

In this formulation of the problem, it is not only the coasting times that minimizes the given cost but also the level of uncertainty that one can include in order to remain in the allowable trade-off range.

Finally, real-time issues have not been investigated nor discussed in the scope of this research. In fact the motion planning problem has been addressed through the use of optimal control techniques, which suffer from high computationnal costs and numerical issues that

make them unsuitable for many real-time applications. Furthermore, t he optimizer was chosen in order to ensure the best optimal solutions and is therefore highly sophisticated. Its implementation in real-time will be very time-consuming and therefore costly. There are ways to implement this methodology in real-time as investigated in [24]. However, this research only focuses on the methodology itself rather than its real-time implementation.

APPENDIX A

ANALYSIS OF MECHANICAL CONTROL SYSTEMS ON LIE GROUPS

A.1 Mathematical Preliminaries

A.1.1 Differential Geometry

Following [8], let Q be the configuration manifold or space such that any element $q \in Q$ defines the configuration of the object. A local coordinate chart $\mathcal{X} : Q \to \mathbb{R}^n$ is defined assuming that Q is locally diffeomorphic to $\mathbb{R}^n$. We will further define the *tangent space* T_qQ to Q at the configuration q whose elements are *tangent vectors*. The *tangent bundle* TQ is defined such that $TQ = \{\bigcup_{q\in Q} T_qQ\,,\, \dim(TQ) = 2n\}$. The *cotangent space* is denoted as $T_q^*Q = \{\bigcup T_q \,|\, T_q \to \mathbb{R} \text{ is a linear function}\}$ and the *cotangent bundle* is designated as $T^*Q = \{\bigcup_{q\in Q} T_q^*Q\}$. To add more clarity, $\boldsymbol{v}$ denoting a velocity vector will be an element of the tangent space T_qQ while a covector $\boldsymbol{p}$ denoting the momentum will belong to the cotangent bundle T^*Q and finally vector forces $\boldsymbol{f}$ will be elements of the cotangent space T_q^*Q.

A.1.2 On Lie Groups

A Lie algebra $\mathfrak{g}$ is a vector space equipped with a skew-symmetric, bilinear bracket $[\cdot\,;\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ that satisfies the Jacobi identity

$$[u;[v;w]] + [v;[w;u]] + [w;[u;v]] = 0$$

For the purpose of this research we are interested in the definition of particular matrix Lie groups, which are the rotation group and its associated Lie algebra and the group of rigid displacements with its corresponding Lie algebra. We will restrict our definitions to $\mathbb{R}^2$

Rotation Group The rotation group is an orthonormal group which is defined as

$$SO(2) \quad \triangleq \quad \{R \in \mathbb{R}^{2\times 2} | R^T R = I, \det(R) = 1\} \tag{10}$$

The associated Lie algebra $\mathfrak{so}(2)$ is the space of skew symmetric matrices

$$\mathfrak{so}(2) \triangleq \{S \in \mathbb{R}^{2\times 2} | S^T = -S\} \tag{11}$$

Rigid Displacements The Euclidean group of rigid displacements, that is rotations and translations on $\mathbb{R}^2$ is denoted as $SE(2)$

$$SE(2) \triangleq \left\{ \begin{bmatrix} R & b \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} | R \in SO(2), b \in \mathbb{R}^2 \right\} \tag{12}$$

The associated Lie algebra $\mathfrak{se}(2)$ is defined as

$$\mathfrak{se}(2) \triangleq \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \tag{13}$$

A.2 Mechanical Control Systems

The reader is referred to consult [9, 8, 21, 10] for full information on Lie algebra and Lie group frameworks. However we will provide preliminaries on the Lie group framework that was used throughout this research.

A.2.1 Simple Mechanical Control Systems on Lie Groups

Given a mechanical system, it is desired to determine the position of its components in an inertial reference frame. We are interested in simple mechanical control systems whose lagrangian is equal to the kinetic minus the potential energy. The definition of mechanical controls systems on Lie groups can be found in [12, 8]

A.3 Planar Rigid Body Example

Consider, for example, a hovercraft gliding over a body of water without friction. The dynamics of the model can be described by the rigid body equations as presented in [21]. Using the notation from [8, 12] let $g(t) = (\psi, x, y)$ be the configuration of the system where the pair (x, y) describes the position of the vehicle and ψ describes its orientation (heading) on a plane relative to the inertial frame. For the planar motion of the vehicle R is the

direction cosine matrix

$$R = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$$

$b \in \mathbb{R}^2$ is a translation vector. We define $g(t) \in SE(2)$ to be the rigid body transformation that maps a body fixed orthonormal frame into an inertial frame so that $g(t)$ describes the position at time t of the vehicle in the plane and its orientation at time t with respect to the inertially fixed axis. Using equations 10 and 11, g can be expressed as

$$g = \begin{bmatrix} \cos\psi & -\sin\psi & x \\ \sin\psi & \cos\psi & y \\ 0 & 0 & 1 \end{bmatrix}$$

similarly, using equations 12 and 13,

$$\hat{\xi} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}$$

where w and v are the angular and linear velocities in body axes.

Given a constant rigid body motion g_0, a constant turning rate $\dot{\psi}$, the general form of the evaluation of the configuration of a planar rigid body is given by

$$g(t) = \begin{bmatrix} \cos\dot{\psi}t & -\sin\dot{\psi}t & v_x/\dot{\psi} \\ \sin\dot{\psi}t & \cos\dot{\psi}t & v_y/\dot{\psi} \\ 0 & 0 & 1 \end{bmatrix} g_0 \tag{14}$$

where v_x and v_y are the components of the velocity vector v in body axes.

A.4 Maneuver Automaton Equations

The evolution of a dynamic system while in trim and maneuver using the maneuver automaton vocabulary was defined as

$$\boldsymbol{x}_{2k} = \phi(\boldsymbol{w}_{q_k}\tau_k)\,\boldsymbol{x}_{2k-1} \qquad \text{in a trim condition} \qquad k = 1, ..., n_T\,; \tag{15a}$$

$$\boldsymbol{x}_{2k+1} = \boldsymbol{x}_{2k} + \Delta\boldsymbol{x}_p \qquad \text{in a maneuver} \qquad k = 1, ..., n_M\,. \tag{15b}$$

in absence of uncertainties and

$$\boldsymbol{x}_{2k} = \phi((\boldsymbol{w}_{q_k} + \epsilon_{\boldsymbol{w}_{q_k}})\tau_k)\,\boldsymbol{x}_{2k-1} \qquad \text{in a trim condition} \qquad k = 1, ..., n_T\,; \tag{16a}$$

$$\boldsymbol{x}_{2k+1} = \boldsymbol{x}_{2k} + \Delta\boldsymbol{x}_p + \epsilon_{\Delta\boldsymbol{x}_p} \qquad \text{in a maneuver} \qquad k = 1, ..., n_M\,. \tag{16b}$$

accounting for the uncertainties. The equations of motion for maneuver conditions are trivial to obtain since they are characterized by a fixed displacement and heading change. We will therefore concentrate our attention on trim conditions in the absence of uncertainties. The evolution of the system is given by

$$\boldsymbol{x}_{2k} = \phi(\boldsymbol{w}_{q_k}\tau_k)\,\boldsymbol{x}_{2k-1} \qquad k = 1, ..., n_T$$

Before the evolution of the system while in a trim condition is defined in more detail, we will illustrate the different transformations that were used. Let the position and orientation

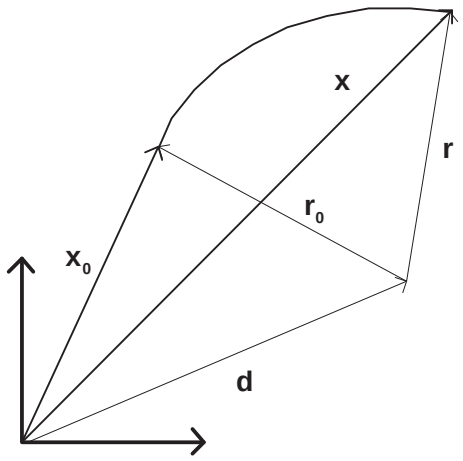


Figure 59: Transformations in Inertial Frame

relative to the inertial frame of an initial point be denoted as $\boldsymbol{x}_0$ and $\boldsymbol{\alpha}_0$ respectively (see Fig. 59). Further, let $\bar{\boldsymbol{v}} = (v_1, v_2, 0)$ and $\bar{\boldsymbol{w}} = (0, 0, w_3)$ be the body attached velocity and angular velocity components characterizing the trim condition. Consequently, the velocity and angular velocity of the initial point relative to the inertial frame can be expressed as $\boldsymbol{v}_0 = \alpha_0\bar{\boldsymbol{v}}$ and $\boldsymbol{w}_0 = \alpha_0\bar{\boldsymbol{w}}$ respectively. Furthermore, the direction cosine matrix, relative to

a rotation around the z axis (recall the formulation of the angular velocity) is defined as

$$R(\bar{\boldsymbol{w}}t) = \begin{bmatrix} \cos w_3 t & -\sin w_3 t & 0 \\ \sin w_3 t & \cos w_3 t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $t \in \mathbb{R}^+$. We are interested in the evolution of the system from the initial position and orientation $\boldsymbol{x}_0$ and $\boldsymbol{\alpha}_0$ to an arbitrary final position and orientation $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ which is reached after t seconds. From Fig. 59,

$$\begin{aligned} \boldsymbol{x}_0 &= \boldsymbol{r}_0 + \boldsymbol{d} \\ \boldsymbol{x} &= \boldsymbol{r} + \boldsymbol{d} \\ \boldsymbol{r} &= R(\bar{\boldsymbol{w}}t)\boldsymbol{r}_0 \\ \boldsymbol{\alpha} &= R(\bar{\boldsymbol{w}}t)\boldsymbol{\alpha}_0 \\ \boldsymbol{v} &= R(\bar{\boldsymbol{w}}t)\boldsymbol{v}_0 \end{aligned}$$

which yields

$$\begin{aligned} \boldsymbol{x} &= R(\bar{\boldsymbol{w}}t)\boldsymbol{r}_0 + \boldsymbol{x}_0 - \boldsymbol{r}_0 \\ \boldsymbol{r}_0 &= \frac{1}{w_3} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{v}_0 \end{aligned}$$

which is reformulated as

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{\alpha}_0 \begin{bmatrix} \sin w_3 t & \cos w_3 t - 1 & 0 \\ 1 - \cos w_3 t & \sin w_3 t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1/w_3 \\ v_2/w_3 \\ 0 \end{bmatrix} \tag{17}$$

Now, using the notation introduced in this work let A_{2k-1} be the initial orientation of the vehicle once it starts trimming. Since the vehicle will remain in a relative equilibria q_k for τ_k seconds, the final orientation of the vehicle can be expressed, as a function of τ_k as

$$A_{2k} = \begin{bmatrix} \cos \dot{\psi}_k \tau_k & -\sin \dot{\psi}_k \tau_k \\ \sin \dot{\psi}_k \tau_k & \cos \dot{\psi}_k \tau_k \end{bmatrix} A_{2k-1}$$

The position, relative to the inertial frame of the vehicle is obtained using Eq. 17

$$\boldsymbol{x}_{2k} = \boldsymbol{x}_{2k-1} + A_{2k-1} \begin{bmatrix} \sin\dot{\psi}_k\tau_k & \cos\dot{\psi}_k\tau_k - 1 \\ 1 - \cos\dot{\psi}_k\tau_k & \sin\dot{\psi}_k\tau_k \end{bmatrix} \begin{bmatrix} \frac{u_{\text{fwd}_k}}{\dot{\psi}_k} \\ \frac{u_{\text{side}_k}}{\dot{\psi}_k} \end{bmatrix} \tag{18}$$

APPENDIX B

ON WORST CASE SCENARIO APPROACH

We are interested in investigating the effect of uncertainties on the overall behavior of the system. It must be pointed out that the same scaling factor is used, throughout the optimization for all the uncertainties in all primitives. This means that uncertainties are either added or subtracted from the motion primitives for a given sequence. However, it is in general difficult to predict the effect of uncertainties on motion primitives and hence on the overall behavior of the system. By restricting γ to take the values -1, 0, or 1 we are assuming that we can predict the effect of the uncertainties on the system. We will present in detail how we can construct the library such that this assumption becomes a valid one. In order to do this, we will first study a simple example.

B.1 Simple Example

Let us consider a very simple Maneuver Automaton Representation of 2 forward flight trims and 2 connecting maneuvers. The trim and maneuver data that was used is provided in Table 27 and Table 28 respectively. It is desired to reach the final position $x_f = [50; 0]$.

Table 27: Simple Problem: Trim Library

q	u_{fwd}	u_{side}	r	$\epsilon_{u_{\text{fwd}}}$	$\epsilon_{u_{\text{side}}}$	ϵ_r
1	1	0	0	0.1	0.1	0.0
2	4	0	0	0.8	0.8	0.0

Table 28: Simple Problem: Maneuver Library

p	q_{from}	q_{to}	$\Delta \boldsymbol{x}_{p_x}$	$\Delta \boldsymbol{x}_{p_y}$	$\Delta\psi$	$\epsilon_{\Delta \boldsymbol{x}_{px}}$	$\epsilon_{\Delta \boldsymbol{x}_{py}}$	$\epsilon_{\Delta\psi}$	T	ϵ_T
1	2	1	5.0	0.0	0.0	-0.5	0.0	0.0	2.0	-0.5
2	1	2	5.0	0.0	0.0	-0.5	0.0	0.0	2.0	-0.5

The initial heading is zero such that the system flies straight. The problem is again to find the best sequence of motion primitives that will allow the vehicle to reach the final

position within set bounds. The system is restrained to start at a slow forward flight which is trim 2. With a maximum trim depth of 2 for a given sequence, there are 2 possible trajectories: Now if we allow the freedom, at each motion primitive, that the uncertainties

Table 29: Possible Sequences

q	p	q
1		
1	2	2

can act any way without restriction, the uncertainty combinations presented in Table 30 will be obtained.

Table 30: Uncertainty Scenarios

Sequence			Uncertainty Scenario		
1			-1		
			0		
			1		
1	2	2	-1	-1	-1
			-1	-1	0
			-1	-1	1
			-1	0	-1
			-1	0	0
			-1	0	1
			-1	1	-1
			-1	1	0
			-1	1	1
			0	-1	-1
			0	-1	0
			0	-1	1
			0	0	-1
			1	1	1
			0	0	1
			0	1	-1
			0	1	0
			0	1	1
			1	-1	-1
			1	-1	0
			1	-1	1
			1	0	-1
			1	0	0
			1	0	1
			1	1	-1
			1	1	0
			0	0	0

A solution is considered feasible if all of the uncertainty scenarios in Table 30 represent a feasible solution to the optimization problem.

B.2 Optimization Results

The results obtained are provided in Table 31 and Table 33.

Table 31: Solutions to sequence 1

γ	Cost (time [s])
-1	50.0000
0	45.4545
1	50.0000

Table 32: Solutions to sequence 2

γ			Cost (time [s])
-1	-1	-1	22.7273
-1	-1	0	13.6250
-1	-1	1	10.1724
-1	0	-1	22.4545
-1	0	0	13.2500
-1	0	1	9.7586
-1	1	-1	22.1818
-1	1	0	12.8750
-1	1	1	9.3448
0	-1	-1	22.7273
0	-1	0	13.6250
0	-1	1	10.1724
0	0	-1	20.1818
1	1	1	9.3448
0	0	1	9.7586
0	1	-1	22.1818
0	1	0	12.8750
0	1	1	9.3448
1	-1	-1	22.7273
1	-1	0	13.6250
1	-1	1	10.1724
1	0	-1	22.4545
1	0	0	13.2500
1	0	1	9.7586
1	1	-1	22.1818
1	1	0	11.6250
0	0	0	12.0000

Table 33: Solutions to sequence 2

From this table it can be seen that the lowest cost corresponds to the uncertainty scenario where the scaling factor γ multiplying all uncertainties is 1 for all motion primitives. Similarly, the maximum cost is achieved for all uncertainties multiplied by γ = -1. This provides a preliminary example in backing up our motivation in using only one scaling factor for all uncertainties, instead of testing each uncertainty scenario possible for each trim. Also, as can be seen from Table 33, the number of optimal control problems to be run and compared is not 3 anymore but is exponentially growing as the number of possible uncertainty choices at each motion primitive is 3. As such, for a given sequence depth of $n_T + n_M$, n_T being the number of trajectory trims and n_M being the number of trajectory maneuvers, the number, N_{opt} of optimal control problems to be run for only one sequence of primitives becomes

$$N_{opt} = (n_T + n_M)^3$$

For the purposes of this research, this will correspond to a computationally infeasible problem and therefore it is in our interest to justify the use of only one scaling factor for each primitive. Recall that trim conditions are characterized by their body axes velocities,

- u_{fwd} forward velocity with associated uncertainty $\epsilon_{u_{fwd}}$
- u_{side} side velocity with associated uncertainty $\epsilon_{u_{side}}$
- r yaw rate with associated uncertainty ϵ_r

Uncertainties are added to the velocities as follows

$$u^r_{fwd} = u_{fwd} + \gamma\,\epsilon_{u_{fwd}} \tag{19a}$$

$$u^r_{side} = u_{side} + \gamma\,\epsilon_{u_{side}} \tag{19b}$$

$$r^r = r + \gamma\,\epsilon_r \tag{19c}$$

where the superscript "r" indicates the real values. Now looking back to the trim library provided in Table 27, one can see that $u_{fwd} > 0$, $u_{side} = 0$ and $\epsilon_{u_{fwd}} > 0$, $\epsilon_{u_{side}} = 0$. It follows that, as uncertainties are added to the trim parameters, i.e. $\gamma = 1$, their overall effect on trajectory trims is such that the vehicle flies faster. This will result on the vehicle

reaching the target in less time, and therefore yields the decrease of the objective function. Similarly, choosing $\gamma = -1$ will result in a decrease in forward speed and hence an increase of the objective function. For the same global overall effect, the uncertainty on the yaw rate ϵ_r will be signed positive for a right turn and negative for a left turn as $r > 0$ for a right turn and $r < 0$ for a left turn. That way, as uncertainties are added according to equation 19c, $\gamma = 1$ will correspond to an increase in magnitude of the yaw rate. Recapitulating,

- $\gamma = 1$ corresponds to uncertainties acting on the system such that it flies faster and reaches the goal in lesser time;
- $\gamma = 0$ corresponds to the case when no uncertainty is acting on the system;
- $\gamma = -1$ corresponds to the uncertainties acting on the system such that it flies slower and hence reaches the goal in more time.

We would like to obtain the same homogeneous effect of uncertainties while maneuvering. Recall that maneuvers were characterized by

- finite transition time T and associated uncertainty ϵ_T
- fixed displacement $\Delta \boldsymbol{x}_p$ and associated uncertainty $\epsilon_{\Delta \boldsymbol{x}_p}$
- fixed heading change $\Delta\psi$ and associated uncertainty $\epsilon_{\Delta\psi}$

Uncertainties affect maneuver parameters as follows

$$T^r = T + \gamma\, \epsilon_T \tag{20a}$$

$$\Delta \boldsymbol{x}_p^r = \Delta \boldsymbol{x}_p + \gamma\, \epsilon_{\Delta \boldsymbol{x}_p} \tag{20b}$$

$$\Delta\psi^r = \Delta\psi + \gamma\, \epsilon_{\Delta\psi} \tag{20c}$$

Let us first concentrate our study on the uncertainty associated with the maneuver duration. If the sign on ϵ_T is positive, then $\gamma = 1$ will correspond to an increase in the maneuver duration, which, in turn, will engender the increase of the objective function. As such, we will have the exact contrary effect than what we have gotten through our analysis with trims. As a remedy, we will store the maneuver duration error ϵ_T as negative. That

way, as $\gamma = 1$, using equation 20a, the maneuver duration with the effect of uncertainties becomes less then the maneuver duration as stored in the library. Similarly, if one spends less time in a maneuver, it is also assumed that one will cover less distance and therefore, the displacement change due to the maneuver must be less while uncertainties are present. Therefore,

$$\Delta \boldsymbol{x}_p > 0 \Leftrightarrow \epsilon_{\Delta \boldsymbol{x}_p} < 0$$

$$\Delta \boldsymbol{x}_p < 0 \Leftrightarrow \epsilon_{\Delta \boldsymbol{x}_p} > 0$$

Since the maneuver displacement, as presented in the maneuver library, is measured in body axis, $\Delta \boldsymbol{x}_p > 0$ will always be satisfied. It follows that $\epsilon_{\Delta \boldsymbol{x}_p} < 0$. Similarly, $\gamma = 1$ will correspond to the case where the uncertainty on the heading change will cause the heading to decrease, since the required maneuver displacement with the addition of uncertainties is less, in magnitude then the original $\Delta \boldsymbol{x}_p$. Let us also point out, that the heading change required to pass from a steady straight flight to a steady right turn , it be fast or slow, as well as the uncertainty $\epsilon_{\Delta\psi}$ associated, is signed negative. Respectively, heading changes and associated uncertainties relative to transitions connecting steady straight flights to left turns are signed positive.

Choosing the appropriate sign for characteristics of trims and maneuvers within the library of motion primitives allows us to obtain the same "global overall effect" relative to the addition of uncertainties to the problem. It can therefore be predicted that by choosing $\gamma = 1$ as the scaling factor that multiplies each uncertainty added to the characteristics of motion primitives, the vehicle will reach the final target in lesser time. By the same token, choosing $\gamma = -1$, the uncertainties will have the contrary effect on the behavior of the vehicle, which will accomplish the mission in larger time. This justifies the choice of the optimization problem name "Worst Case Scenario Approach", as the worst time scenario is captured due to the proper construction of the library of motion primitives.

B.3 Proof

Let us consider the time invariant mechanical system denoted as S. Let X be the state space of S as depicted in Chapter 5. The behavior, over time, of the system in the absence

of uncertainties satisfies the differential equations on the states $\boldsymbol{x} \in X$ defined as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$$

where $\boldsymbol{u}$ denotes control inputs to the system. Those control inputs are not known a priori to the system.

Let us consider again the same mechanical system S whose states' behavior over time with the addition of uncertainties is determined by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{\tau}, \gamma)$$

where $(\boldsymbol{q}, \boldsymbol{\tau})$ represents the hybrid control input to the system and γ is the scaling factor multiplying the uncertainties in the systems dynamics. The state vector is given by:

$$\boldsymbol{x}(t) = \Phi(\boldsymbol{x}, \boldsymbol{q}, \gamma)\boldsymbol{x}(t_0)$$

Locally, assuming that perturbations are small enough and linearizing, we can rewrite $\boldsymbol{\Phi}(\boldsymbol{x}, \gamma)$ as

$$\Phi(\boldsymbol{x}, \gamma) \quad = \quad \Phi(\boldsymbol{x}_0, \gamma_0) + \Phi_{\boldsymbol{x}}(\boldsymbol{x}_0, \gamma_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \Phi_\gamma(\boldsymbol{x}_0, \gamma_0)(\gamma - \gamma_0) \; + \mathcal{O}(|(\boldsymbol{x}, \gamma)|)$$

Then, along the trajectory, taking $\boldsymbol{x} = \boldsymbol{x}_0$ and $\gamma_0 = 0$ leads

$$\Phi(\boldsymbol{x}, \gamma) \quad = \quad \Phi(\boldsymbol{x}, 0) + \Phi_\gamma(\boldsymbol{x}, 0)\,\gamma \; + \mathcal{O}(|(\boldsymbol{x}, \gamma)|)$$

hence,

$$\begin{aligned} \Phi(\boldsymbol{x}, -1) &= \Phi(\boldsymbol{x}, 0) - \Phi_\gamma(\boldsymbol{x}, 0) \\ \Phi(\boldsymbol{x}, 1) &= \Phi(\boldsymbol{x}, 0) + \Phi_\gamma(\boldsymbol{x}, 0) \end{aligned}$$

Since $-1 \leq \gamma \leq 1$,

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, -1) = \Phi_\gamma(\boldsymbol{x}, 0)\,(\gamma + 1) \tag{21}$$

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, 1) = \Phi_\gamma(\boldsymbol{x}, 0)\,(\gamma - 1) \tag{22}$$

If $\Phi_\gamma(\boldsymbol{x}, 0) < 0$, equations (21,22) satisfy

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, -1) < 0$$

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, 1) > 0$$

which equivalently yields

$$\Phi(\boldsymbol{x}, 1) < \Phi(\boldsymbol{x}, \gamma) < \Phi(\boldsymbol{x}, -1)$$

Now if $\Phi_\gamma(\boldsymbol{x}, 0) < 0$, equations (21,22) satisfy

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, -1) > 0$$

$$\Phi(\boldsymbol{x}, \gamma) - \Phi(\boldsymbol{x}, 1) < 0$$

and thus

$$\Phi(\boldsymbol{x}, -1) < \Phi(\boldsymbol{x}, \gamma) < \Phi(\boldsymbol{x}, 1)$$

We conclude that $\forall \boldsymbol{x} \in X$, we have

$$\Phi(\boldsymbol{x}, 1) < \Phi(\boldsymbol{x}, \gamma) < \Phi(\boldsymbol{x}, -1) \qquad \text{or} \tag{23}$$

$$\Phi(\boldsymbol{x}, -1) < \Phi(\boldsymbol{x}, \gamma) < \Phi(\boldsymbol{x}, 1) \tag{24}$$

which implies, for $-1 \leq \gamma \leq 1$

$$\text{traj}\left(\boldsymbol{x}, \Phi(\boldsymbol{x}, \gamma)\right) \subset \text{Interior}[(\boldsymbol{x}, \Phi(\boldsymbol{x}, 1)), (\boldsymbol{x}, \Phi(\boldsymbol{x}, -1))] \tag{25}$$

REFERENCES

[1] Alur, R., Courcoubetis, C., Halbwacks, N., Henzinger, T. A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S., "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, pp. 3–34, 1995.

[2] Alur, R., Courcoubetis, C., Henzinger, T. A., and Ho, P.-H., "Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems," *Hybrid Systems I, Lecture Notes in Computer Science*, vol. 736, pp. 209–229, 1993.

[3] Arthur E. Bryson, J. and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation and Control*. Hemisphere Publishing Corperation, 1975.

[4] Betts, J. T., "Survey of numerical methods for trajectory optimization," *AIAA Journal of Guidance, Control and Dynamics*, vol. 21, 1998.

[5] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*. Advances in Design and Control, 2001.

[6] Bottasso, C. L., Croce, A., Leonello, D., and Riviello, L., "Optimization of critical trajectories for rotorcraft vehicles," *Journal of the American Helicopter Society*, 2004. Accepted for publication.

[7] Branicky, M. S., Borkar, V. S., and Mitter, S. K., "A unified framework for hybrid control: Background, model, and theory," Tech. Rep. LIDS-P-2239, Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1994.

[8] Bullo, F., *Nonlinear Control of Mechanical Systems: A Riemannian Geometry Approach*. PhD thesis, Pasadena, California, 1998.

[9] Bullo, F. and Leonard, N. E., "Motion control for underactuated mechanical systems on lie groups," *Proceedings of the European Control Conference*, p. 480.

[10] Bullo, F., Leonard, N. E., and Lewis, A. D., "Controllability and motion algorithms for underactuated lagrangian systems on lie groups," *IEEE Transactions on Automatic Control*.

[11] Faiz, N., Agarwal, S., and Murray, R., "Differentially flat systems with inequality constraints: An approach to real-time feasible trajectory generation," *AIAA Journal of Guidance, Dynamics and Control*, vol. 24, no. 2, pp. 219–227, 2001.

[12] Frazzoli, E., *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, Cambridge, Massachusetts, 2001.

[13] Frazzoli, E., "Quasi-random algorithms for real-time spacecraft motion planning and coordination," *Acta Astronautica*, vol. 54, pp. 485–495, 2003.

[14] Frazzoli, E., "Maneuver based motion planning for nonlinear systems with symmetries," *Submitted to AIAA Journal of Guidance, Dynamics and Control*, 2004.

[15] Frazzoli, E., Dahleh, M., and Feron, E., "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," *IEEE Transaction on Systems, Man and Cybernatics*, 1999.

[16] Frazzoli, E., Dahleh, M., and Feron, E., "Real time motion planning for agile autonomous vehicles," *AIAA Journal of Guidance, Dynamics and Control*, vol. 25, no. 1, 2002.

[17] Henzinger, T. A., "The theory of hybrid automata," *Verification of Digital and Hybrid Systems*, 2000.

[18] Hsu, D., Kindel, R., Latombe, J.-C., and Rock, S., "Randomized motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[19] Johnson, E. N. and Kannan, S. K., "Adaptive trajectory control for autonomous helicopters," *AIAA Journal of Guidance, Control, and Dynamics*. Accepted for publication.

[20] LaValle, S. M. and Kuffner, J., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[21] Murray, R. M., Li, Z., and Sastry, S. S., *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[22] Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S., "An approach to the description and analysis of hybrid systems," *Hybrid Systems I, Lecture Notes in Computer Science*, vol. 736, pp. 149–178, 1993.

[23] Piedmonte, M. and Feron, E., "Aggressive maneuvering of autonomous aerial vehicles: A human centered approach," *International Symposium on Robotics Research*.

[24] Schouwenaars, T., Mettler, B., Feron, E., and How, J. P., "Robust motion planning using a maneuver automaton with build-in uncertainties," *American Control Conference*, 2003.

[25] Stengel, R., "Towards intelligent flight control," *IEEE Transaction on Systems, Man and Cybernatics*, vol. 23, pp. 1699–1717, May 1993.